

Zachariah Steventon-Barnes

Registration number 100386904

2025

**Declaring Total War On The (Web
Notifications) Pusherman:
Educating People About The Risks Of Web
Push Notification Scams**

Supervised by Debbie Taylor



University of East Anglia
Faculty of Science
School of Computing Sciences

Abstract

Web Push Notifications (WPNs) are a mechanism for web pages to display notifications to users. Existing research and my personal experience indicate that they are frequently misused by scammers. This project also surveyed users and found that permission prompts to allow them and popups displaying them are also commonly misunderstood by users. However, as each web browser has a standard prompt, they offer a promising opportunity for user education.

This project set out to test the hypothesis that “an educational quiz website can increase users’ knowledge of web push notification scams in a manner that is accessible and which they find usable”. To do this, I conducted a survey to identify poorly understood aspects of WPNs. Based on this, I created an example of such an educational quiz website. Finally, I conducted user testing in which users were asked a set of questions about WPNs before and after using the quiz website.

Users performed 52% better on the survey questions after using the quiz website, their confidence in correct answers increased from a mean of 5.1 to 7.6 (out of 10), and they perceived the quiz website to be usable. However, I encountered substantial accessibility challenges when designing the website for users that have a screen reader. As such I assess the hypothesis as being partially correct. Additional work would be required to develop a system accessible for users of screen readers.

Acknowledgements

I would like to thank Debbie Taylor for her detailed and clear feedback and to Associate Professor Masaya Sato for corresponding with me about his research paper. I would also like to thank all the wonderful 115 people who took the time to participate in my survey. And finally, thank you to Terry for bringing joy and happiness to all of the class of ‘25.

Contents

1. Introduction	7
1.1. Background	7
1.2. Motivation	7
1.3. Aims and Objectives	8
1.4. Structure	8
2. Background	9
2.1. Web Push Notification Technology	9
2.2. Motivating example	9
2.3. Malicious Web Push Notifications	10
2.4. The (In)Effectiveness Of Traditional Security Technologies Against Web Push Notification Malvertising	11
2.5. Novel approaches to Web Push Notification Malvertising	12
2.6. Why scammers pick Web Push Notifications	12
2.7. Potential effectiveness of education	13
2.8. Computer Scams	13
2.9. Social Engineering	14
2.10. Social Engineering Education	15
2.11. Computer System Design	15
2.12. Accessibility	16
3. Survey Preparation and Design	17
3.1. Heading section	17
3.2. Classification section	17
3.3. Data Section	18
3.3.1. Question 1 - identifying scam CAPTCHAs	18
3.3.2. Question 2 - understanding WPN permission prompts	19
3.3.3. Question 3 - identifying the origins of malicious WPNs	19
3.3.4. Question 4 - blocking malicious notifications part 1	20
3.3.5. Question 5 - blocking malicious notifications part 2	20
3.3.6. Question 6 - blocking all notifications	20
3.4. Scores	20
3.5. Ethics Approval	20

4. Survey Results	21
4.1. Participants and demographics	21
4.2. Results by demographic	21
4.2.1. By Age and Gender	21
4.2.2. By Education	22
4.2.3. By heard of WPNs and self-assessed digital literacy	22
4.3. Results by question	22
4.3.1. Question 1 - identifying scam CAPTCHAs	22
4.3.2. Question 2 - understanding WPN permission prompts	23
4.3.3. Question 3 - identifying the origins of malicious WPNs	23
4.3.4. Question 4 - blocking malicious notifications part 1	23
4.3.5. Question 5 - blocking malicious notifications part 2	24
4.3.6. Question 6 - blocking all notifications	24
4.4. Overall conclusions	24
5. Development Preparation	25
5.1. Development methodology	25
5.2. Persona	25
5.3. Requirements title list	26
5.4. Visual design	26
5.5. UML state machine	27
5.6. Question answering	27
6. Development Implementation	28
6.1. Quiz	28
6.2. Database design	28
6.3. Security	29
6.4. Question structure	29
6.5. User settings structure	29
6.6. Attempts structure	30
6.7. Evaluation against MoSCoW	30
6.8. Design and Accessibility	30
7. User Testing Preparation	31
7.1. Overview	31

7.2. Testing increased knowledge and usability	31
7.3. Gathering qualitative data	31
7.4. Ethics approval	31
8. User Testing Results	32
8.1. Usability	32
8.2. Educational results	32
8.2.1. Group scores	32
8.2.2. Individual scores	33
8.2.3. Question 5 - blocking malicious notifications part 2	33
8.2.4. Question 6 - blocking all notifications	34
8.3. Positive and Negative Qualitative Feedback	34
8.4. Additional Qualitative Information	34
9. Discussion and Evaluation of Objectives	35
9.1. Objective 1 - Understand WPNs, scams, and scam education	35
9.2. Objective 2 - Understand usability and accessibility	35
9.3. Objective 3 - Survey to identify misunderstandings	35
9.4. Objective 4 - Design quiz to improve knowledge	35
9.5. Objective 5 - Develop quiz to improve knowledge	36
9.6. Objective 6 - User testing to measure knowledge increase	36
10. Conclusion and Future Work	37
10.1. Overall Finding	37
10.2. Additional Findings and Contributions	37
10.3. Future Improvements To The Educational Quiz Website	37
10.4. Future Work	38
References	38
A. Graphs for survey results	44
B. Web Push Notifications Network Diagram	55
C. Web Push Notifications Sequence Diagram	55

D. Mid-Fi Prototype Of Homepage	56
E. Mid-Fi Prototype Of Question	56
F. Textual Use Case Question	57
G. Flow Chart Question	58
H. Dependencies	58
I. Evaluation against MoSCoW requirements titles	59
J. Database Excerpt	59
K. Password Hashing Method	60
L. Pug Question Block	60
M. Testing Results From Question 5	60
N. Testing Results From Question 6	61
O. User Feedback Quotes	61
P. Positive Qualitative Feedback	62
Q. Negative Qualitative Feedback	62
R. Glossary	62

1. Introduction

1.1. Background

New web technologies often come with new security threats (Carboneri et al., 2023). One area that is an ongoing research topic (Sato and Mandai, 2024) and for which counter-measures are insufficient (Matsuzaki and Sato, 2025) is Web Push Notifications (WPNs).

WPNs allow websites to display notifications even when the website is closed, generally appearing as system notifications (Carboneri et al., 2023) i.e. outside of the browser window (Pantelakis et al., 2022). These notifications combine 3 browser APIs: Push API, Service Workers, and Notifications API (Carboneri et al., 2023).

WPNs are frequently malicious. Subramani et al. (2020) finds that of 21,541 WPN messages collected, 5,143 were used for advertising, and 2,615 were identified as malicious (12% of all WPNs, and 51% of WPN advertisements). This compares to 0.23% of regular advertisements (Confiat, 2020) meaning WPN advertisements are 200 times more likely to be malicious. Pantelakis et al. (2022) indicates that the proportion of WPNs used for advertising has risen.

1.2. Motivation

“If ‘Slick’ Willie Sutton, the oft-quoted 1930s gentleman bank robber, was embarking on his criminal career today he would, likely as not, be contemplating using the internet to commit his robberies: ‘[b]ecause that’s where the money [now] is!’” ~ Wall (2007)

However, Wall (2024) states that what he believes has changed is that Sutton’s successors “would no longer be planning arduous and dangerous million-dollar bank robberies because . . . networked technologies enable him to commit millions of micro-robberies [robberies individually too small to investigate] with much less personal risk”.

The criminals seem to agree. Scams and computer misuse are a large and growing share of crime in the UK (ONS, 2024). 67% of crimes against a person and 48% of all crimes were categorised as fraud or computer misuse (up from 55% and 35% in 2018).

I experienced this first hand with both sets of grandparents accidentally allowing malicious WPNs which they were unable to identify the origins of, leaving them scared and confused.

However, as WPNs require acceptance through a standardised permission prompt

and display in a standardised manner they should be particularly easy to tackle through education. This project set out to test that hypothesis.

1.3. Aims and Objectives

As stated in Steventon-Barnes (2024), the main aim of this project was to test the hypothesis that “an educational quiz website can increase users’ knowledge of web push notification scams in a manner that is accessible and which they find usable”.

With this aim in mind, I intend to achieve the following objectives:

1. Develop an understanding of web push notifications, scams, web push notification scams, and educating users about scams.
2. Develop an understanding of designing usable and accessible systems.
3. Carry out a user survey to identify common misunderstandings around web push notification scams.
4. Design an educational quiz website to improve users’ knowledge of web push notifications, especially in areas identified by the survey as being weak.
5. Develop an educational quiz website to improve users’ knowledge of web push notifications.
6. Complete user testing to identify whether this quiz increases users’ knowledge.

1.4. Structure

The remainder of this report will be structured as follows: Section 2 will provide a background of the relevant topics, Section 3 will describe the design of the survey with Section 4 detailing the results. Section 5 will discuss the resulting design of the quiz with Section 6 discussing the results of development. Section 7 will describe the process of user testing with testing results in Section 8. Section 9 will describe the main findings and evaluate the work done against the aims and objectives. Finally, Section 10 will conclude the portfolio and suggest future research topics.

A glossary of relevant terms is included in Appendix R.

2. Background

2.1. Web Push Notification Technology

Web Push Notifications combine three different APIs: the Push, Service Worker, and Notifications APIs (Carboneri et al., 2023; MDN, 2023b). To send a push message, a website first needs to create a service worker which requests permission to send notifications (Carboneri et al., 2023; Emathingier, 2015; Gaunt, 2016), which the Push API standard states requires the “express permission” of the user (W3C, 2024). The website then receives a PushSubscription which contains a URL for the servers of a “Push Service” to which it sends its push messages (Carboneri et al., 2023; Emathingier, 2015; Gaunt, 2016). An overview of how these APIs connect can be seen in Figure 1 and a network diagram for a WPN can be seen in Appendix B.

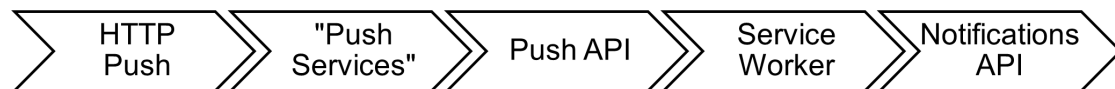


Figure 1: Overview of APIs used to send WPNs

To send a push message, the website’s server then connects to this URL over the HTTP Push protocol which delivers the push message over that Push Service (Emathingier, 2015; Gaunt, 2016). A Push event is then sent to the service worker that registered it (Carboneri et al., 2023; Gaunt, 2016) which can in turn display it using the Notifications API standard (Carboneri et al., 2023; W3C, 2024; MDN, 2023a). A sequence diagram illustrating this can be seen in Appendix C.

When sent, web push notifications generally appear as system notifications and can appear even when the web app that set them up is not running (Carboneri et al., 2023; Subramani et al., 2020) and can appear when the browser isn’t running on mobile browsers and desktop Safari (CanIUse, 2024).

2.2. Motivating example

In 2023, my grandfather – who considers himself “fairly computer savvy” having been an electrical engineer – was attempting to purchase tickets to see tennis at Wimbledon. He browsed to a malicious site and clicked ‘allow notifications’. A day later, his computer started displaying unwanted notifications advertising products, telling him he had

won online prizes, and telling him his system had security issues. These all had a strange URL at the bottom of the message, though he was unaware that this was supposed to indicate they were WPNs.

He believed a virus had infected his system and ran a scan with McAfee anti-virus which did not detect any viruses (the reasons for which are discussed in Section 2.4). He then ran a Windows Defender Offline Scan after which the notifications disappeared. Exactly why an Offline Scan should cause malicious Web Push Notifications to disappear is unclear, however when I inspected his browser, it did not list any sites with the notifications permission granted, possibly indicating it is (or was) part of an Offline Scan to revoke all notifications permission in Microsoft Edge.

2.3. Malicious Web Push Notifications

Subramani et al. (2022)'s literature review categorises web push notification attacks into two categories: continuous execution and Push API and Notifications Abuse. Continuous execution attacks zombify computers into joining a botnet where they can be used for whatever purpose the bot herder desires – Papadopoulos et al. (2019) suggests cryptocurrency mining, password cracking, and DDoS attacks while Wall (2024) mentions that cybercriminals also sell on botnets.

However, despite considerable research into the possibility of such an attack and related attacks (e.g. Chinprutthiwong et al. (2021); Komodo Research (2019); Lee et al. (2018); Pan et al. (2016); Papadopoulos et al. (2019); Rushanan et al. (2016)), I have been unable to find any cases of a WPN botnet spotted in the wild.

This is likely because, while the Push API in theory allows the delivery of push messages without displaying a notification, in practice this is restricted by browsers (Lee et al., 2018; Subramani et al., 2022). Subramani et al. (2022)'s Push API and Notifications Abuse category contains attacks where displaying notifications is integral, which they subdivide into phishing, malvertising, and stalkerware .

Regarding stalkerware, Carboneri et al. (2023); Chinprutthiwong et al. (2021); Lee et al. (2018) propose a variety of attacks on flawed WPN implementations compromising user privacy. However, they are out of scope as education can do little to stop them.

I believe Subramani et al. (2022)'s separation of phishing and malvertising is not helpful as phishing is often the ultimate aim of malvertising. Further their source for phishing – Lee et al. (2018) – provides an example of malvertising whose end goal is to

get users to install a Chrome extension as opposed to phishing.

Subramani et al. (2020) shows that, in contrast to WPN botnets, WPN malvertising is both real and prevalent. They record that of 21,541 WPN messages collected, 5,143 were used for advertising, and 2,615 were identified as malicious (12% of all Web Push Notifications, and 51% of WPN advertisements). For comparison, 0.23% of advertisements in general in Q1 2020 were malicious according to Confiant's Ad Quality Index (Confiant, 2020). While the studies use different methods, if both are accurate WPN delivered ads are 200 times more likely to be malicious than online advertisements in general.

While focused on in-page WPNs, Pantelakis et al. (2022) counts all service workers communicating with an ad server at 3,289 out of 7,444 indicating the proportion of WPNs used for advertising has risen to 44%.

2.4. The (In)Effectiveness Of Traditional Security Technologies Against Web Push Notification Malvertising

Subramani et al. (2020) found adblockers Easylist Blocklist, AdGuard, and Adblock-Plus were all unable to block a single malicious WPN service worker and only Easylist was able to block any requests from service workers and then only 132 out of 8031 (1.6%), though they indicate this may be the result of a bug in Chromium at that time.

However, even after the bugs are fixed, Pantelakis et al. (2022) points out traditional adblocking techniques would not be able to prevent WPN delivered ads.

According to a case study of one WPN malvertising campaign in a white paper from cybersecurity firm Indelible (Angiolelli, 2020), only 1 out of 4 anti-malware solutions detected it and there was a low detection rate using proxy or DNS based malicious website identification. Meanwhile, user account control does not prevent it as it only affects the specific user's account.

Anti-malware vendors repeat this with Arntz (2019) stating WPNs escape Malwarebytes scans, though since 2022 Malwarebytes has had a category for WPNs implying it can now detect WPNs (Malwarebytes, 2022).

2.5. Novel approaches to Web Push Notification Malvertising

Due to the ineffectiveness of traditional security measures and the lack of existing research on this topic, new measures are an ongoing research topic (Harbach et al., 2024; Sato and Mandai, 2024). Subramani et al. (2020) found it did not affect a single website of a random 300 website sample studied.

Google is redesigning its UX for WPNs to reduce nuisance prompts. Bilogrevic et al. (2021) describes the start but notes that they do not directly tackle deceptive WPNs, deliberately and successfully did not change permission grant rates, and have yet to design “an easy and more obvious escape hatch” for changing permission status.

Harbach et al. (2024) continues this pattern, though notes that in December 2022 Google reset notification permissions for the top-30 sites by uninteracted-with notifications. Notably Bilogrevic et al. (2021) and Harbach et al. (2024)’s user based activation mechanisms target frequent users consequently excluding those likeliest to have the lowest digital literacy. Further, the roll out remains “partial” as of Harbach et al. (2024).

Further it is unclear this quieter prompt will actually help reduce malicious WPNs. Edge deployed it for all users (Richards et al., 2020) however both motivating examples discussed in Section 1.2 were using Edge after the roll out.

Sato and Mandai (2024) propose using a keyword list to identify malicious WPNs. However according to Sato (2025), they were only able to process 7 of 128 WPN, receiving 5 true positives, 1 false positive, and 1 false negative. This suggests that without improving the number of notifications that can be scanned and the detection rate, this process will be ineffective.

Sato and Mandai (2024) suggest it can be improved using text similarity analysis or combined with their image similarity analysis, Matsuzaki and Sato (2025).

2.6. Why scammers pick Web Push Notifications

There are a number of reasons I believe scammers choose web push notifications:

Firstly, as mentioned above, security technologies have limited effectiveness against malicious Web Push Notifications.

Secondly, the rate of granting permissions for web push notifications is almost certainly higher than the rate of installing software. Bilogrevic et al. (2021) finds that for a traditional Chrome notifications prompt, 12% of permission requests were granted on desktop and 23% on mobile.

Further, c. 8% of desktop clients approved every notification request they saw while that number was c. 15% on mobile. However, Bilogrevic et al. (2021) only covered 10 days with c. 80% of desktops and c. 90% of mobiles seeing less than 5 prompts.

Thirdly, WPNs have many of the same benefits to scammers that they do to legitimate users. They can be used on different browsers and different operating systems (Lee et al., 2018). Push API and Notifications API are currently supported on all major desktop and mobile browsers, though they are only supported in Safari iOS when the website is saved to the home screen and Push API is only supported on Safari MacOS Ventura and later (MDN, 2024a,b).

Fourthly, WPNs are not easily logged or processed by analysis tools, making incident response difficult for organisations (Lo, 2021).

2.7. Potential effectiveness of education

While I have been unable to find any research specifically on educating users on the risks of WPNs, a common response to malicious WPNs in both cybersecurity industry publications and technology news has been to advise users on how to monitor them (Arntz, 2019; Krebs, 2020). Lee et al. (2018) also recommends that users should be aware of the phishing risks posed by WPNs.

I believe that educating users on push notifications is likely to be particularly effective as W3C (2024) dictates that these notifications can only be received after being approved and this approval is handled by browsers through a standardised dialogue giving scammers limited opportunity to manipulate them. They can also be easily disabled once the user knows how to identify them and where to disable them.

2.8. Computer Scams

Scams and computer misuse make up a large and growing share of crime in the UK (ONS, 2024). 67% of reported incidences of crimes against a person and 48% of all crimes were categorised as fraud or computer misuse. These numbers are up from 55% and 35% respectively in 2018 and are likely similar everywhere with high rates of computer use.

This makes tackling these crimes a key challenge to both developers of computer systems and anyone trying to make the UK a safer place. Assessing and tackling the

use of computer technologies to conduct these crimes is therefore a key objective of the computer science profession.

Malicious Web Push Notifications generally exist as part of various forms of virtual scams. Wall (2024) defines virtual scams by writing, “whereas virtual stings tend to focus upon the business community, virtual scams are primarily aimed at victimizing individual users. A virtual scam, in contrast to a virtual sting, is designed to tempt anybody who is exposed to it.”

The form mentioned most frequently in literature on malicious WPNs (Angiolelli, 2020; Subramani et al., 2020) are scareware scams. Wall (2024) describes scareware scams as sending users messages that are designed to convince users that their computer is infected with malware and to pay the scammer to have it removed.

2.9. Social Engineering

Whereas scam describes the end goal of malicious WPNs, social engineering is the technique that they use to accomplish it. Hadnagy (2018) defines Social Engineering as “any act that influences a person to take an action that may or may not be in his or her best interests.”

As such, I would argue malicious WPN ads contain 2 acts of social engineering, separated by time. First, convincing the user to allow notifications and second, the ultimate scam – be that a scareware scam or a lottery scam. On the other hand, WPN based botnets contain 1 act of social engineering, convincing the user to allow notifications.

This is not entirely clear, as what constitutes social engineering is somewhat unclear. Tetri and Vuorinen (2013) and Wang et al. (2020) identify a difficulty in the literature in defining social engineering, or as Wang et al. (2020) puts it, “the conceptual intension (connotation) of social engineering is not consistent.”

Another issue is that literature on social engineering focuses on attacks on organisations to the point where it is unclear that attacks against individuals qualify. Wall (2024) writes “social engineers tend to use their knowledge of systems to make themselves appear credible to staff”. By contrast Gururaj et al. (2024) explicitly acknowledged the existence of social engineering attacks targeting individuals.

A third issue with deciding whether or not to categorise this as a social engineering attack is the focus in literature on tailored one-on-one attacks. Again, Gururaj et al. (2024) includes automated attacks in its study of social engineering.

2.10. Social Engineering Education

Regardless of whether this fits the common consensus of what social engineering is, I believe that it is sufficiently applicable that research into educating users to tackle social engineering will help address the issues WPN malvertising presents.

Social Engineering Education strategies are often categorised into traditional and modern or traditional and next-generation (referred to here as 1st generation and 2nd generation) (Aldawood and Skinner, 2019, 2018; Gardner and Thomas, 2014). 1st generation tools are based on passive learning techniques whereas 2nd generation tools use interactive learning techniques (Gardner and Thomas, 2014).

1st generation techniques are characterised as being tedious and forgettable, hard to apply (Aldawood and Skinner, 2019), and subject to “message malaise” (Gardner and Thomas, 2014). Schneier (2013) suggests that security training was so flawed as to be a waste of money, and it would be better spent designing systems that do not allow poor security behaviours (like choosing a weak password) or provide well explained warning messages.

This view is not shared by most other cybersecurity professionals (Gardner and Thomas, 2014) or papers written on the subject (Aldawood and Skinner, 2018), though as Gardner and Thomas (2014) put it about the state of training in 2014, “everybody agrees we have to do something”.

The result was 2nd generation tools which use methods such as: video based delivery, gamification, self-paced delivery methods, simulations, and virtual labs. The literature indicates that professionals believe that these tools are more effective and their popularity is increasing (Aldawood and Skinner, 2019, 2018; Gardner and Thomas, 2014).

However, I was unable to find any research showing that 2nd generation tools result in better knowledge retention than 1st generation tools, or even that they result in any knowledge retention at all. Of Aldawood and Skinner (2018)’s examples: Holdsworth and Apeh (2017) surveyed participants only before the training; Jin et al. (2018) surveyed participants only after the training; and Beckers and Pape (2016) is not intended as a training tool at all but rather a requirements elicitation tool.

2.11. Computer System Design

Two key sources on the design of computer systems are Nielsen (2024) which provides a list of 10 heuristics computer systems can be designed against and Norman (2013)

who provides 7 general design principles.

Norman (2013)'s 1st principle is discoverability, that it should be possible to identify the current system state and possible actions. Visibility of system status is Nielsen (2024)'s 1st heuristic and his 6th heuristic states that elements, actions, and options should be visible so users can recognise them as opposed to have to recall them.

Norman (2013)'s 2nd principle is feedback, which is that the system tells the user about the results of actions. This aligns again with Nielsen (2024)'s 1st heuristic with his 9th heuristic adding that clear error messages are important.

Norman (2013)'s 4th principle is that affordances exist which allow desired actions. Nielsen (2024)'s 7th heuristic states this should include short cuts for experienced users while his 3rd heuristic states it should provide an easy undo function.

Norman (2013)'s 5th principle states that there should be good signifiers to aid the discoverability of affordances. Nielsen (2024)'s 8th heuristic however cautions that rarely needed information should be removed to avoid cluttering interfaces.

Norman (2013)'s 6th principle is that there should be a good mapping between the controls and their actions using their spatial distribution to link them. Nielsen (2024)'s 2nd heuristic similarly adds computer systems should match the real world.

Norman (2013)'s 7th principle is to provide constraints to guide user actions. Nielsen (2024)'s 5th heuristic suggests that systems should eliminate or make it hard to access error prone states.

Norman (2013) states these principles aid discoverability while his 3rd principle is to help users form a clear conceptual model. While a conceptual model does not allow a user to understand the system's current state or available actions, it allows them to predict the results of the actions that they take on the system's state.

2.12. Accessibility

WCAG 2.2 is a well-recognised set of guidelines for accessibility, referenced by both law and used in literature. WAI (2024) states that WCAG 2.2 is based on 4 main principles, that web content should be: perceivable – users should be able to sense the content being displayed; operable – users should be able to control the content being displayed; understandable – users should be able to comprehend the content being displayed and how to control it; and robust – content should be interpreted reliably by a range of user agents including assistive technology.

3. Survey Preparation and Design

Having completed my literature review, I discovered that browser permission prompts and settings were the only effective defence against malicious WPNs. I next conducted a survey as listed in objective 3 to discover how well users understood them.

The survey consisted of the three sections listed by Eva et al. (2020): heading, classification, and data. The heading and classification sections were formatted as single Microsoft Form sections to help respondents complete the form faster.

The data section questions were each placed into their own Form section as some questions provide the answers to preceeding questions and this approach discourages participants from changing their previous answers to be correct.

Particular care was given to keep the survey brief as Eva et al. (2020) notes that the length of a survey can be one of the key causes of participants choosing not to answer or providing bad answers. I also activated the progress bar in Forms and numbered the questions out of 6 to encourage users to complete the survey.

3.1. Heading section

The heading section was used for the project information and consent form as required by UEA's ethics policy.

3.2. Classification section

The classification section asks users their age, gender, and education level. The sources analysed (Lloyds, 2024; Mancino, 2023; ONS, 2019) indicated a large difference in digital skills by age and education level and a small one by gender.

I chose to ask for the highest level of education started rather than achieved as I largely distributed the survey to people I know, many of whom are computer science undergraduates. Doing otherwise may have led to a misleading result that respondents who only have A-levels have a higher level of digital literacy than those with degrees.

Lloyds (2024) indicates that there is also a divide by employment status and by disability. However, it is unclear if this remains the case once that data is adjusted for age. Lloyds (2024) also indicates that there is a difference by social grade and income level. However, I expect to struggle to get participants of a range of social grades. My income data would also likely be skewed by a preponderance of university students.

I decided not to ask any further questions of specific demographics with the aim of keeping the survey brief. I asked two questions specifically about digital literacy. The first simply asked participants to score their digital literacy. This uses a Microsoft Forms NPS® question (Li, 2018) as it allows the words ‘very low’ and ‘very high’ to be displayed at opposite ends of the scale, but it requires the score to be on a 0 to 10 scale which users may find too long. A 1 to 7 score was considered, however as the scale had to be explained above the question, I considered it to be more potentially confusing.

The second simply asks users whether they had previously heard the term ‘Web Push Notifications’ with the answers of “Yes”, “No”, and “Maybe”.

All the questions in the classification section are optional, and the general demographics ones have a “prefer not to say” option.

3.3. Data Section

The data section consisted of a series of multiple choice questions with questions 1 to 4 being factual questions with a correct and a series of incorrect options to assess knowledge of different areas of WPN scams and are accompanied by a follow up question asking how confident participants feel in their answers. Questions 5 and 6 ask whether participants can carry out certain tasks with the answers of “Yes”, “No”, and “Maybe”.

I have marked questions 1 to 4 as required as I do not want users to leave them uncompleted because they do not feel confident in their answers. It is important to understand what actions users would take even if they are uncertain of them.

I have tried to ensure participants do not quit the survey on the grounds that they do not know the answers to the questions by asking them what they ‘think’ the answer is, by reminding them after each question, “Do not worry if you do not know the answer, just select whichever one you think is most likely and indicate below how confident you feel in your answer.”

3.3.1. Question 1 - identifying scam CAPTCHAs

The first question asks participants “Which of the following is not a legitimate CAPTCHA (tool to check you are human)?” It is accompanied by an image of a Web Push Notification scam masquerading as a CAPTCHA, similar to figure 4 of (Angiolelli, 2020) and images of reCAPTCHA v2, hCaptcha, and Cloudflare Turnstile.

3.3.2. Question 2 - understanding WPN permission prompts

The second question shows an image of a WPN scam which pretends to require WPNs to be allowed to play a video. The first potential answer I gave was that the scam was legitimate.

The next two answers were that it would allow the website to store a cookie and that it would allow the website to get the geographic location of the browser. I choose these answers as users often visit websites with allow/deny dialogues for these two questions and I wanted to understand if users confused WPN dialogues for those dialogues.

The next three answers asked about various potential ways of displaying notifications. I provided two incorrect answers with the word ‘notifications’ in them so that users did not select the correct answer just because it was the only one that mentioned ‘notifications’.

Importantly, both the incorrect answers specify that notifications will only be shown when the website is open – a difference in functionality not explained in the permission dialogue pictured, which I felt was important.

3.3.3. Question 3 - identifying the origins of malicious WPNs

The third question shows a picture of a scam notification pretending to be from McAfee delivered through the Chrome web browser from the (fictitious) website *notifier-server.example.com* (based off *notify-service[.]com*). The first potential answer again was the pretence the scammers set, that it was from McAfee.

The second was that it was from Google Chrome. Discussions before I started this project with people who had allowed WPN scams have indicated to me that the appearance of the browser’s name on the notification is often misunderstood or even makes the scam seem more legitimate to users (one told me that they thought the scammers were very clever placing the ‘e’ on their notifications to make them seem more legitimate because it was the Microsoft logo – it is the logo for Microsoft Edge through which the scammers were sending the notifications).

The third asks whether it is a virus installed on their system. Discussions indicate that WPN scams are often mistaken for traditional viruses causing users to respond by attempting to remove them through virus scans which are generally ineffective as opposed to removing them through the web browser’s settings as the makers of web browsers intended.

The 4th and 5th answers correctly identify the website notifier-server.example.com as the source and state that they are being delivered through Chrome. They differ as the 4th answer states that the website is open while the 5th states that it may not be open.

3.3.4. Question 4 - blocking malicious notifications part 1

Question 4 confirms that the image shown in question 3 is a scam notification and asks users how they think they should remove it with option 1 being an antivirus scan, option 2 being disabling all notifications from Chrome in Windows' settings, and option 3 being disabling notifications from notifier-server.example.com in Chrome's settings.

One thing that I examined when I finished collecting responses is whether a misunderstanding in question 3 that the notification is from a virus leads people into incorrectly carrying out an anti-virus scan.

3.3.5. Question 5 - blocking malicious notifications part 2

Question 5 asks participants whether they know how to carry out the option they picked in question 4 with the potential answers of 'yes', 'no', and 'maybe'. I chose to do it like this as opposed to specifying the 'correct' answer so users would not be tempted to go back to question 4 and change their answer.

3.3.6. Question 6 - blocking all notifications

Question 6 is similar to question 5 except it asks participants whether they know how to turn off all notifications from websites.

3.4. Scores

One way I analysed the survey results was by calculating a score from 0 to 4 representing how many of the 4 factual questions (questions 1 to 4) the user had got correct. Questions 5 and 6 are evaluated separately.

3.5. Ethics Approval

The survey received ethics approval from UEA's Faculty of Science Research Ethics Subcommittee as ethics approval ID ETH2425-1059.

4. Survey Results

I distributed the survey to people I knew by e-mail and WhatsApp and on social media through LinkedIn. Some of my friends also distributed it through social media platforms including Facebook and Instagram. One person distributed it to their coworkers through their email saying that it was a good awareness tool as they were going through the process of ISO 27001 certification.

The consequence of distributing to my contacts is that respondents were likely from Western, Educated, Industrialised, Rich, and Democratic (WEIRD) countries (almost exclusively the UK). This likely skewed my results to more knowledgeable individuals.

All Figures showing results in this section can be found in Appendix A.

4.1. Participants and demographics

I received responses from 115 participants. These participants were generally young (median age group 24-34) and tended to be men (62% versus 37% women). They tended to have at least started a university degree (68%) and very few only had GCSEs (5%). They entered a mean score of 7.1 on their self-assessed digital literacy.

74% stated that they had heard of Web Push Notifications before (versus 14% who had not). This was an unexpectedly high result, and may indicate that most participants were mistaken. However the answer to this question did have a substantial impact on the scores (see section 4.2.3).

4.2. Results by demographic

4.2.1. By Age and Gender

Interestingly, examining the 4 factual questions, there is no clear trend by age group as shown in Figure 8. Indeed, the mean for the middle age groups, 25-54, slightly outperforms (mean 2.2) the youngest age group, 18-24 (mean 2). Men slightly outperformed (mean 2.1) women (mean 1.9) as shown in Figure 9. The best performing age and gender group (see Figure 10) was men aged 45-54 (mean 2.6) which is likely because of a bias towards computer professionals within the survey sample for that demographic.

In terms of the average confidence for these questions (see Figure 11), men tended to be more confident (mean 6.1) than women (mean 4.9). However, for almost all age

and gender demographics, the mean confidence was the same regardless of whether the answer was correct or incorrect, with only men aged 45-54 bucking this trend.

4.2.2. By Education

The results for education (see Figure 12) were roughly as expected. People who had GCSEs or no formal education had a mean result of 1.1 while those who started A-levels (2.0) and university degrees (2.2) performed better. People with GCSEs had lower confidence (mean of 4.75 out of 10) than A-levels (5.83) and degrees (5.76).

There was again little difference in confidence (see Figure 13) between correct and incorrect answers within the group. People who had started A-levels were 0.51 points more likely to get the correct answer and 0.61 for those who started university degrees.

4.2.3. By heard of WPNs and self-assessed digital literacy

The results by WPNs were as expected (see Figure 14). People who had heard of WPNs had a better mean score (2.3) than people who hadn't (1.3) or weren't sure (1.4). This was reflected in mean confidence scores of 6.1, 4.2, and 4.9 respectively (see Figure 15).

Results by digital literacy were roughly as expected but contained some surprises. No participants entered a digital literacy below 3. I grouped scores into sets of 2 (see Figure 16). The 3-4 group actually performed the same as the 5-6 group (mean 1.3).

There is a strong correlation between self-assessed digital literacy and confidence both in correct and incorrect answers as shown in Figures 17 and 18. All groups were more confident in correct than incorrect answers with those entering 7-8 having the highest difference (average 0.6).

4.3. Results by question

4.3.1. Question 1 - identifying scam CAPTCHAs

Question 1 received one of the highest proportions of correct answers (see Figure 20) at 69 correct (60%). As confidence increased, the proportion of correct answers also increased (see Figure 21), with 22 people giving the correct answer with >8 confidence compared with 3 people giving an incorrect answer. The most common incorrect answer was Cloudflare's Turnstile, likely because it appears in dark mode.

4.3.2. Question 2 - understanding WPN permission prompts

Question 2 had a low proportion of correct answers (see Figure 22), 49 correct (43%). However, in contrast with the previous question, people generally correctly identified that the permission request was nefarious with only 5 respondents (4%) stating it had no ulterior purpose. This may be due to heightened awareness in the context of the survey.

Again, as confidence increased so did the proportion of correct answers (see Figure 23) with 12 of 19 (63%) being correct for confidence above 8.

The most common incorrect answer (23 responses, 20%) was that it was a cookie permission prompt. It is worrying that people are unable to distinguish between a dialogue from the website and one from the browser and suggests low user understanding of the Chrome website permissions system. Actually, users confusing another browser permission dialogue (geolocation) was far less common (6 responses, 5%).

4.3.3. Question 3 - identifying the origins of malicious WPNs

Question 3 had the highest proportion of correct answers (see Figure 24), 70 (61%). 17 out of 26 (65%) of responses with confidence above 8 were correct (see Figure 25) – indicating confidence here does not indicate an answer is more likely to be correct. The proportion who believed it was legitimate was also low, 6 (5%) believing it had come from McAfee and 6 from Chrome. This indicates that most people in the sample can identify that a malicious WPN is malicious.

The most common incorrect answer (20 responses, 17%) was that it was a website but that the website had to be currently open, a belief which would likely lead users to be confused if they could not find a tab with it open. 13 respondents (11%) believed it was a virus on their system.

4.3.4. Question 4 - blocking malicious notifications part 1

Question 4 had the lowest proportion of correct answers (i.e. dealt with as intended by browser vendors) as shown in Figure 26; 46 (40%). 15 out of 28 (54%) of respondents with confidence above 8 were correct (see Figure 27).

People were equally likely to attempt an anti-virus scan (46 respondents). This is unlikely to stop the malicious WPNs (see section 2.4). Some respondents (23, 20%) said they would disable all notifications from Chrome in their Windows settings. While

this would stop malicious WPNs, it would also disable legitimate WPNs, which may not be the user's intention. This suggests that while people are able to correctly identify a malicious WPN (see question 3) they do not know what to do about it.

As suspected, an incorrect identification of the origin of a malicious WPN led to an increased likelihood of tackling it incorrectly (see Figure 28). 53% (7 out of 13) people who said they believed that it originated from a virus stated they would run an anti-virus scan. This compares with 33% (23 out of 70) who correctly identified the origin.

4.3.5. Question 5 - blocking malicious notifications part 2

I looked at the responses to this question from people who stated that they would disable the WPNs in Chrome's settings as shown in Figure 29. Of these, 27 (59%) stated that they did know how to do it compared to only 7 (15%) who definitely didn't know. This suggests that if people are able to identify a WPN, and know to look in their browser's settings, they believe they can use them to disable it.

4.3.6. Question 6 - blocking all notifications

Similarly, 60 respondents (52%) stated they knew how to disable all WPNs (see Figure 19), compared to 36 (31%) who stated they did not know. Together, questions 5 and 6 indicate people are confident about their browser's settings if they are able to identify the origins of a malicious WPN.

4.4. Overall conclusions

Notification permission prompts (see sections 4.3.1 and 4.3.2), the origins of notifications themselves (see section 4.3.3), and how to disable them (see section 4.3.4) were consistently misunderstood by circa 40 – 60% of users. By contrast, once users did know where notifications came from, and they knew what to do, they felt confident they knew how to do it with only circa 15 – 30% saying they definitely didn't know (see sections 4.3.5 and 4.3.6).

As a result, I decided to focus the quiz mostly on understanding notification prompts and notifications themselves and avoided giving users too much information about how to specifically disable notifications in their browser (which would be challenging anyway as the specific layout of browser settings differ between browsers).

5. Development Preparation

Informed by my survey's findings I began the process of designing the web based educational quiz website as listed in objective 4.

5.1. Development methodology

As a one person team developing a small system and as I wished to collect further survey data and literature during development, I chose to use an agile methodology using a mix of tools from feature driven development and kanban.

5.2. Persona

For the initial development (Steventon-Barnes, 2024), I created a persona of a potential user (Figure 2) loosely based on my motivating examples discussed in Section 1.2.

Maggie Greene

Age: 71
Occupation: Retired
Status: Married
Location: Woodbridge
Digital literacy: Limited



Life Goals:

- See her great-grandson through his childhood
- Ensure the local cub scout branch she is a leader at is successful and enjoyable

Pain-points:

- Feels worried about online scams and computer viruses
- Has poor eyesight making reading text on screen difficult – she uses a screen magnifier, but it doesn't work for all websites

“Whenever I go online I see all these pop-ups and I don't know what they mean.”

Behaviours:

- Uses her tablet to browse the web and read e-mails when at home
- Doesn't use technology on the go because she is worried about scams

Personality:

- Is a leader at her local cub scout branch
- Enjoys participating in outdoor activities
- Enjoys spending time with her friends and family
- Likes how Ed Sheran has “stuck to his Suffolk roots”

- #### Motivations for using the quiz:
- Understand what website pop-ups asking for permission to send notifications mean



Figure 2: A persona of an example user of the system

5.3. Requirements title list

Cadle (2014) suggests that for agile projects a requirements list is most appropriate as they aim to maximise time spent coding and because requirements are potentially changeable. I created a requirements title list (Steventon-Barnes, 2024), separating them using a MoSCoW prioritisation framework and categorising them into functional (FR) and non-functional (NFR) requirements. This can be seen in Figure 3.

Must have <i>(Components required by system)</i> 1.Allow users to answer multi-choice questions (FR) 2.Show users text for questions (NFR) 3.Tell users if their answer was correct (FR) 4.Give users question specific feedback (FR)	Could have <i>(Optional components that are desirable)</i> 8.Show users example web push notifications (FR) 9.Allow users to change settings to better suit visual impairments and dyslexia (NFR) 10.Allow questions to be edited by editing a JSON file (FR) 11.Allow images to be zoomed in on (NFR)
Should have <i>(Components very desirable but may be removed if they require excessive work)</i> 5.Show users images for questions (NFR) 6.Be compliant with WCAG 2.2 at AA level (NFR) 7.Allow users to view previous scores (FR)	Won't have <i>(Desirable components considered too much work for the initial version)</i> 12.A mechanism to save user progress (FR) 13.A mechanism to send questions specific to the browser (FR)

Figure 3: A visual MoSCoW for the system

5.4. Visual design

I created two greyscale prototypes: one of the home screen (Appendix D) and one of a screen showing a question (Appendix E) to assist in the design work for Steventon-Barnes (2024).

As mentioned in Steventon-Barnes (2024), consideration was given to the font stack and colour.

I chose to implement a wide range of configuration options for the visual design including background text colour, foreground text colour, font, and font size as this is considered useful for people with dyslexia (Gupta et al., 2021). During user testing, several participants mentioned that they struggled with the default colour scheme emphasising that allowing users to change it is an important accessibility characteristic (see Section 8.3).

I chose the default font stack of: Century Gothic, Apple Gothic, Didact Gothic (from Google Fonts), Verdana, Sans-Serif

I chose geometric fonts like Century Gothic, Apple Gothic, and Didact Gothic to both help give the website a modern appearance and because they are thought to be easier to learn and read for people with dyslexia (Gupta et al., 2021; BDA, 2023) or limited experience of printed fonts. Didact Gothic was designed for elementary schools and improving literacy rates (Vanyashin, 2016). This accords with my personal experience with autistic spectrum disorder that geometric fonts are easier to cope with.

I also considered the background colour. In line with BDA (2023) I used a light colour background with dark text, which also makes the website seem less bland. Further, it avoids or alleviates issues dark mode causes around eye focus, ‘haloing’ and excessive contrast (Cheddar, 2019).

5.5. UML state machine

As the website has a limited number of pages, I felt it would be appropriate to model it as a UML state machine. A UML state machine shows how the state (in this case page) changes depending on input (in this case the user’s clicks) (Sommerville, 2015). The diagram can be seen in Figure 4.

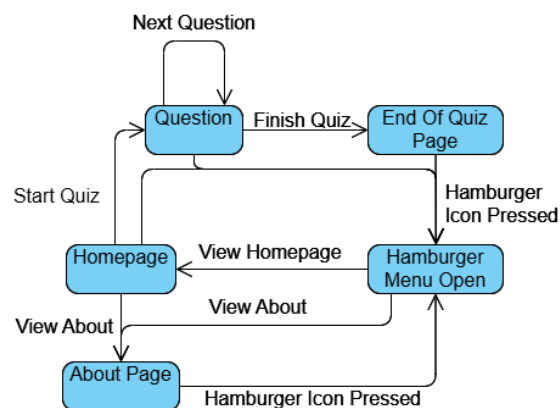


Figure 4: A UML state machine showing navigating the website

5.6. Question answering

For further detail on how the quiz process would work, I created a textual use case for the process which can be found in Appendix F. I also created a flow chart (see Appendix G) to show this in a visual manner.

6. Development Implementation

My final code architecture used a Node.JS and Express webserver using the pug template library and a PostgreSQL database because of my experience in that area. I used ECMAScript modules as opposed to CommonJS modules as these are standardised, have better performance and are considered to have a more readable syntax (Ubah, 2024).

The database code resides within a DatabaseAPI class which uses a singleton design pattern to prevent multiple instances being created and exceeding available memory on the webserver or connections from the database.

Pages displayed to the user are all pug templates which use pug's 'extend' keyword to use a common layout.pug file. This file also includes a block of javascript so the same script can be executed on all page loads.

6.1. Quiz

The quiz uses the question.pug and answer.pug files with the relevant question details being passed to them from the questions.json file which allows for questions to be easily added and modified.

In Appendix H, I have shown the software and packages this system depends upon. For ease of maintenance and security, I have endeavoured to keep these to a minimum while avoiding recreating large bodies of existing work. Samaana et al. (2025) found that small NPM libraries with a higher than median ratio between code from other libraries and their code are 4 times more likely to be exposed to a security vulnerability, suggesting that excessive usage of other libraries is a security risk.

6.2. Database design

The database uses PostgreSQL as a wide-column database, using its JSONB types to store both user settings (see Section 6.5) and attempts at the quiz (see Section 6.6). I have shown this in Figure 5. PostgreSQL being an SQL database makes writing queries easier and reduces vendor lock-in compared to a pure wide-column database like MongoDB.

The DatabaseAPI class also included a large number of internal unit tests to ensure that features, including security functionality like authentication, was working correctly.

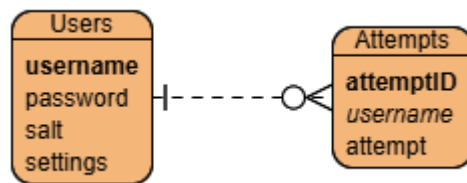


Figure 5: An entity relationship diagram for the database

6.3. Security

Users' passwords are stored according to the OWASP (2025) guidance. I used the Node Crypto library as I wanted to minimise the number of packages. As Node Crypto does not support Argon2, I used Scrypt. See Appendix J for an example of what a password processed like this looks like.

The password and salt columns are up to 200 characters which are designed to enable up to 1024 bit hashes and salts for future proofing and I have taken care to design the Scrypt constants to be clear and easy to edit (see Appendix K).

I used the session-cookie library for session management as it makes scaling easier than a database-based system and because express-session requires additional configuration of the session store.

While I have avoided displaying user-controlled strings, I have also taken advantage of Pug's inbuilt sanitization feature to prevent Cross Site Scripting (XSS) attacks (see Appendix L).

The database uses parameterised SQL queries to prevent SQL injection attacks as well as running on a separate PostgreSQL account to ensure privilege minimisation.

6.4. Question structure

Questions are loaded from a JSON file to make changing them easy. The JSON file is loaded into a constant when the server starts to prevent any risk of path traversal vulnerabilities and improve system performance.

6.5. User settings structure

User settings are stored in a JSON. Long term storage uses the database from which the JSON is loaded into the session cookie at login. This is also designed to improve

website scaling. When settings are changed they are stored to the session cookie and the database.

6.6. Attempts structure

A user's score on each quiz attempt is stored as a quiz attempt JSON containing two values: 'correct' (the number of questions that the user has got correct); and 'wrong' (the number of questions that the user has got incorrect). Dissimilar names were deliberately chosen for these values to minimise the chance of confusion.

This is stored in the session token. An attempt with zero in both scores is stored when the user loads the question 1 page. Every answer page then increments the relevant counter. Finally the results page stores that attempt to the database.

6.7. Evaluation against MoSCoW

I evaluated the system against the MoSCoW requirement titles set out in section 5.3.

As shown in Appendix I, I implemented all 'must have' requirements and all 'should have' requirements that proved possible to implement. I also implemented a good range of 'could have' objectives and provided a promising groundwork for 1 objective I intended for future versions.

6.8. Design and Accessibility

I successfully implemented the ability to change font and background. A version with the default settings can be found in Appendix D while a version with custom settings can be seen in Appendix E, where they can also be compared to the mid-fi prototypes they were based on.

One obvious flaw with this design is that the questions are based on images which are not accessible for users who use screen readers. One option is to provide alternate text but there is no way of ensuring this alternate text matches the way screen readers would represent a notification request.

If I had created example web push notifications these would have been read by screen readers in the same way as malicious WPNs would be. However, the URL parameter, which is the only way users are supposed to be able to identify which website the request came from, would of course only be able to show the address of the quiz website.

7. User Testing Preparation

7.1. Overview

The plan for user testing was to have users undertake the following 3 activities in order:

1. Take the questionnaire (see section 3)
2. Use the quiz website
3. Take a modified version of the questionnaire with the classification section removed (as the classification section from the first survey could be used to analyse user results) and a few questions about the quiz website added.

7.2. Testing increased knowledge and usability

The first component of the hypothesis is whether users have gained knowledge from taking the quiz. I intended to do this by testing whether users: (1) had become more accurate and (2) had become more confident in answers they had already got correct.

The third component of the hypothesis is whether users find the system usable. In the second questionnaire (step 3), users were asked to score “how well did you think the website worked” from 0 (“Extremely unusable”) to 10 (“Extremely usable”). I will consider an average score above 5 to indicate that users found the website usable.

7.3. Gathering qualitative data

I asked users two questions to elicit qualitative data. The first question was “what did you like about the website?” which was intended to elicit positive feedback to ensure aspects that users viewed as positive would not be changed in future revisions.

The second question was “what changes do you think would have improved the website?” which was intended to elicit negative feedback so that future revisions could make improvements to the website.

7.4. Ethics approval

The user testing received ethics approval from UEA’s Faculty of Science Research Ethics Subcommittee as ethics approval ID ETH2425-1710.

8. User Testing Results

8.1. Usability

The mean usability score was 6.1 with a median score of 7 (see Figure 6). This indicates that users found the website slightly more usable than not and that the mean was lowered by a small number of particularly low scores. I attempted to analyse these scores to find a common theme but these scores spanned a wide range of different demographics.

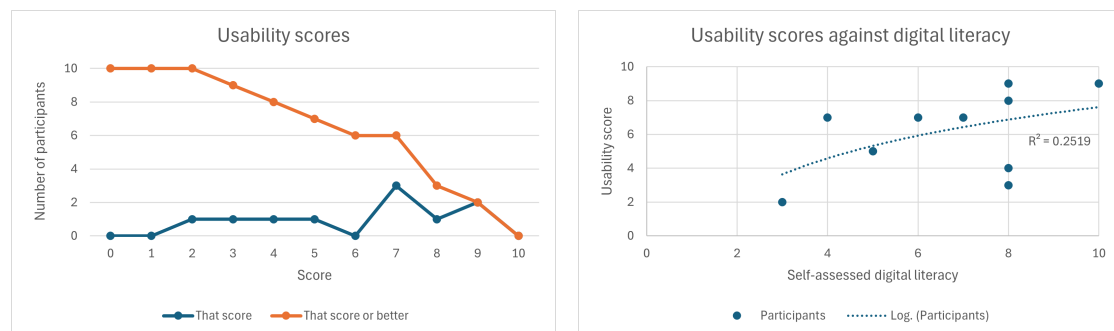


Figure 6: Usability scores frequency and cumulative frequency graph (left) and as a scatter plot compared to self-assessed digital literacy (right)

I also compared usability scores to self-assessed digital literacy (Figure 6). This shows that people with a higher digital literacy found the tool most usable, which suggests that it needs to be made more usable in future revisions.

8.2. Educational results

8.2.1. Group scores

Before using the quiz, the average correct score (see Section 3.4) from the survey was 2.3, the average confidence was 4.3, the average confidence for correct answers was 5.1, and the average confidence for incorrect answers was 3.2.

After using the quiz, the average score from the survey was 3.5 (a 52% improvement), the average confidence was 7.1, the average confidence for correct answers was 7.6, and the average confidence for incorrect answers was 3.4.

This indicates that the quiz makes users more accurate and more confident while having a minimal effect on confidence for incorrect answers.

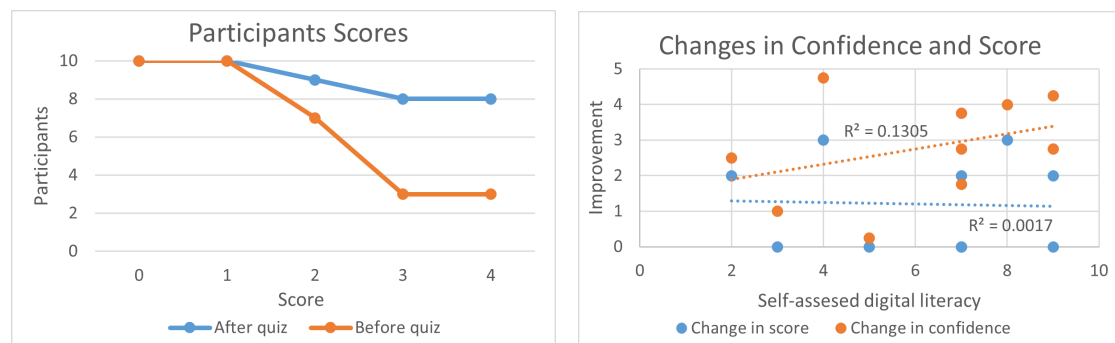


Figure 7: Cumulative frequency graph of scores before and after the test (left) and increase in scores and confidence compared to self-assessed digital literacy (right)

8.2.2. Individual scores

I compared individual participants' first score to their second score. The mean change in correct answers was 1.2 and the median change was 2. For no participants did completing the quiz make them score worse on the questionnaire (see Figure 7).

Accounting for the number of questions participants originally answered incorrectly, the achieved improvement out of the maximum potential improvement was 71%. Interestingly, participants either got all previously incorrect questions correct or got none of the previously incorrect answers correct.

8.2.3. Question 5 - blocking malicious notifications part 2

Question 5 asks participants "do you know how you would carry out your answer to the previous question (question 4)?" The results show a shift from 'yes' to 'maybe' (see Appendix M). People giving the correct answer to question 4 as opposed to an approach they were more familiar with may have resulted in a drop in confidence.

However, looking at the per results data, this only explains 1 result. Another person seems to have simply become less confident in this question due to taking the quiz.

It is hard to develop a quiz question on specifically how to disable notifications in the browser's settings as it differs between browsers, so multiple questions would have to be developed, and each browser directed to the appropriate one.

I chose not to develop such a system (see Section 5.3 Requirement 11) because the user survey indicated that users felt relatively confident in this area (see Section 4.4).

8.2.4. Question 6 - blocking all notifications

This question generally gives a more accurate picture of whether this aspect of the quiz was successful as the intended action is specified in the question and does not change from user to user. The results show a slight increase in confidence (see Appendix N), however again the results were disappointing.

This was again likely because the quiz did not display a browser settings page as it was hard to create one for each browser.

8.3. Positive and Negative Qualitative Feedback

Users appreciated the website being clear ('clear' featured in almost half of responses, see quotes in Appendix O and word cloud in Appendix P) and appreciated its simple layout. They also liked the question specific feedback.

The most recurrent criticism was the choice of colour palate (as can be seen in quotes in Appendix O and word cloud in Appendix Q). My penchant for lilac was apparently not shared by my user test group and when combined with the blue colour of hyperlinks was viewed by one participant as "almost unreadable". Although logged in users are able to change the background colour, it seems prudent to change the default in future.

Other criticisms were often also what one user described as "minor visual display items". Users wanted to be able to view the image alongside the answers, so implementing that is one potential improvement. Higher quality images were one item desired.

Users also desired clearer answers with fewer technical terms. One user suggested that the answers highlighted the relevant area of the image.

8.4. Additional Qualitative Information

One issue that pervaded the user testing was mouse control. My user testing group had a preponderance of MacOS users who found my two button track pad difficult to use. I had plug in mice which they also struggled with. It became clear that users did not understand which website the questions in the 2nd questionnaire were referring to and asked me for clarification as we were completing the test.

It also became clear that users did not pay as much attention to the survey as I had hoped and many users missed the links to high quality versions of the images the survey was based on.

9. Discussion and Evaluation of Objectives

9.1. Objective 1 - Understand WPNs, scams, and scam education

I have completed a comprehensive literature review of WPNs and WPN scams to address objective 1 (see Sections 2.1, 2.3, 2.4, and 2.5). I believe I have reviewed all the relevant literature on this topic including articles published while my research was being conducted (e.g. Matsuzaki and Sato (2025)) as well as corresponding with authors when I required clarification. While additional research into the UX of push notifications may have contributed, it would have been tangential.

I am less satisfied with my literature review into scams and particularly into educating users about scams (see Sections 2.8, 2.9, and 2.10). The literature reviews I examined on this subject (Aldawood and Skinner, 2019, 2018) did not identify any papers demonstrating that 2nd generation tools are superior to 1st generation tools or that 2nd generation tools give any benefit to their users (see Section 2.10). However, there do not seem to be any literature reviews on this subject since 2020.

9.2. Objective 2 - Understand usability and accessibility

I am reasonably pleased with my literature review into how to design usable (see Section 2.11) and accessible (see Section 2.12) systems. I was unable to identify a mechanism for proving the accessibility of the system, however I did develop a good understanding of what makes a system accessible.

9.3. Objective 3 - Survey to identify misunderstandings

I am pleased with my survey results (see Section 4) which allowed me to focus my system on the key areas of missing knowledge (as described in Section 4.4). I am very pleased with the number of participants, however the sample had too few older participants and its level of digital literacy was unrepresentative.

9.4. Objective 4 - Design quiz to improve knowledge

I am pleased with my system design (see Section 5), which I think effectively balanced providing adequate system specifications while also not being excessively detailed in

a way that might have made development inflexible or make the design process overly time consuming.

However, development was hampered by a lack of more technical details. This stems partially from the differing paradigms of Node and UML. Whereas UML is focused on describing Object Oriented Programs, Node operates on a procedural basis. This means Node web servers are hard to describe with UML diagrams as they are ill suited to, for example, class diagrams.

Additionally, issues with the colour scheme (see Section 8.3) were not identified through the prototyping process as the prototypes were created in grey scale.

9.5. Objective 5 - Develop quiz to improve knowledge

The development process was successful and resulted in a system that could be successfully used to test the hypothesis. This is, to my knowledge, the first time an educational quiz for WPNs has been developed. As demonstrated in Appendix I, I delivered well against my MoSCoW. However, it became evident in the development process that due to the limitations of its format an educational quiz website could not be accessible to users who used a screen reader (see Section 6.8).

9.6. Objective 6 - User testing to measure knowledge increase

The user testing was completed successfully allowing me to test my hypothesis. I demonstrated that the tool increases users' levels of knowledge resulting in a 52% increase in correct answers (see Section 8.2) and that they found it usable (see Section 8.1). Given the lack of prior evidence that a 2nd generation tool can increase users' knowledge on a social engineering topic, this appears to be a particularly useful finding.

If conducted again, I would have asked more quantitative questions to assess different aspects of usability, such as "did you enjoy using the website?" and "did you find the website easy to navigate?"

10. Conclusion and Future Work

10.1. Overall Finding

Overall, I found my hypothesis (“an educational quiz website can increase users’ knowledge of web push notification scams in a manner that is accessible and which they find usable”) to be partially true. I found that it did increase users’ knowledge - resulting in a 52% increase in correct answers (see Section 8.2). Their confidence in correct answers increased from a mean of 5.1 to 7.6 (out of 10). I also found that users did find it to be usable (see Section 8.1). However it was not possible to design a website for this purpose in a way that was accessible for users of screen readers (see Section 6.8).

This demonstrates, for what I believe is the first time (see Section 2.10), that a 2nd generation social engineering education tool can increase users’ knowledge. Further, I created, also for what I believe is the first time, a 2nd generation tool to address WPN scams which delivered well against my MoSCoW (see Appendix I).

10.2. Additional Findings and Contributions

I also found that a high proportion of users are unable to distinguish a legitimate CAPTCHA from a WPN scam CAPTCHA (see Section 4.3.1) and that most users do not understand the meaning of the Google Chrome WPN permission prompt (see Section 4.3.2). Further I found that a large proportion of users could not identify the origin of a scam WPN (see Section 4.3.3) and consequently were likely to try to resolve it in ways that are ineffective (see Section 4.3.4).

These findings indicate that the nature of WPNs which merge native like behaviour with a permission prompt that is substantially less noticeable than permission prompts typically required to install an application (e.g. Windows Smart Screen) is confusing for users. This combined with their frequent use for malicious purposes and the inability of traditional security technologies to address them raises the question of whether WPNs as a whole do more harm than good for users.

10.3. Future Improvements To The Educational Quiz Website

One improvement to the website would be to change the default background colour. The adaptable nature means the user settings class is easily changed to do this.

Another change, before adding any settings that would be disruptive if maliciously triggered (such as a delete account button) would be to add Cross Site Request Forgery (XSRF) protections. In particular, when forms are sent to users, a XSRF token should be generated and stored in the database and in a hidden field in the form. Then when the form is returned, the value of the hidden field is compared against the database. This allows the website to check whether the form was generated by the website.

A more substantial change would be to rewrite the application in TypeScript. JavaScript's lack of types in combination with its use handling of untrusted user inputs makes securing applications difficult. For instance, consider when a variable that is supposed to be an integer is passed to the database. In JavaScript the Database cannot guarantee that it is an integer, and must either assume that validation before hand worked correctly or do its own validation, adding code complexity.

10.4. Future Work

One promising way of improving the educational website and adding valuable knowledge to the literature would be to compare different styles of educational website. The user testing methodology I have developed allows websites to be evaluated for the amount of knowledge users have gained and how usable they find such websites.

This could be used to evaluate whether 2nd generation interactive social engineering education methods actually convey a benefit by randomly placing users into one group that uses the existing website and another that uses a static website with just text and images. If the first group outperforms the second, we could conclude 2nd generation tools are more effective. We could also compare the existing website to one that makes more use of 'gamification' to see which 2nd generation tools are most effective. In either case, an increased number of participants would make the experiment more robust.

Another avenue for future work would be to develop a system through a modified web browser or a native application that allows more control over notifications allowing a better understanding for users who have a screen reader.

References

Aldawood, H. and Skinner, G. (2018). Educating and Raising Awareness on Cyber Security Social Engineering: A Literature Review. In *2018 IEEE International Confer-*

- ence on Teaching, Assessment, and Learning for Engineering (TALE), pages 62–68. ISSN: 2470-6698.
- Aldawood, H. and Skinner, G. (2019). Reviewing Cyber Security Social Engineering Training and Awareness Programs—Pitfalls and Ongoing Issues. *Future Internet*, 11(3):73. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.
- Angiolelli, F. (2020). PushBug - Uncovering Widespread Push Notification (RFC 8030) Abuse in the Wild. Technical report, Indelible LLC.
- Arntz, P. (2019). Browser push notifications: a feature asking to be abused | Malwarebytes Labs.
- BDA (2023). Dyslexia friendly style guide.
- Beckers, K. and Pape, S. (2016). A Serious Game for Eliciting Social Engineering Security Requirements. In *2016 IEEE 24th International Requirements Engineering Conference (RE)*, pages 16–25, Beijing, China. IEEE.
- Bilogrevic, I., Engedy, B., Iii, J. L. P., Taft, N., Hasanbega, K., Paseltiner, A., Lee, H. K., Jung, E., Watkins, M., McLachlan, P. J., and James, J. (2021). "Shhh...be {quiet!}" Reducing the Unwanted Interruptions of Notification Permission Prompts on Chrome. pages 769–784.
- Cadle, J. (2014). Requirements Engineering. In *Developing Information Systems: Practical guidance for IT professionals*.
- CanIUse (2024). Web Notifications.
- Carboneri, A., Ghasemisharif, M., Karami, S., and Polakis, J. (2023). When Push Comes to Shove: Empirical Analysis of Web Push Implementations in the Wild. In *Annual Computer Security Applications Conference*, pages 44–55, Austin TX USA. ACM.
- Cheddar (2019). Is Dark Mode Actually Better For Your Eyes? - Cheddar Explains.
- Chinprutthiwong, P., Vardhan, R., Yang, G., Zhang, Y., and Gu, G. (2021). The Service Worker Hiding in Your Browser: The Next Web Attack Target? In *24th International Symposium on Research in Attacks, Intrusions and Defenses*, pages 312–323, San Sebastian Spain. ACM.

- Confiant (2020). Demand Quality Report. Technical Report Q1 2020.
- Emathinger, P. (2015). *Unified Push Messaging for Web Applications*. PhD thesis, University of Applied Sciences Upper Austria.
- Eva, M., Paul, D., and Cadle, J. (2020). Investigating the Business Situation. In *Business Analysis*.
- Gardner, B. and Thomas, V. (2014). *Building an Information Security Awareness Program*. Syngress, Marshall University, Huntington, WV, USA.
- Gaunt, M. (2016). How push works | Articles.
- Gupta, T., Aflatoony, L., and Leonard, L. (2021). “A Helping Hand”: Design and Evaluation of a Reading Assistant Application for Children with Dyslexia. In *33rd Australian Conference on Human-Computer Interaction*, pages 242–251, Melbourne VIC Australia. ACM.
- Gururaj, H. L., Janhavi, V., and Ambika, V. (2024). *Social Engineering in Cybersecurity: Threats and Defenses*. CRC Press.
- Hadnagy, C. (2018). *Social Engineering: The Science of Human Hacking*. Wiley, 2 edition.
- Harbach, M., Bilogrevic, I., Bacis, E., Chen, S., Uppal, R., Paicu, A., Klim, E., Watkins, M., and Engedy, B. (2024). Don’t Interrupt Me - A Large-Scale Study of On-Device Permission Prompt Quieting in Chrome. In *Proceedings 2024 Network and Distributed System Security Symposium*, San Diego, CA, USA. Internet Society.
- Holdsworth, J. and Apeh, E. (2017). An Effective Immersive Cyber Security Awareness Learning Platform for Businesses in the Hospitality Sector. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, pages 111–117.
- Jin, G., Tu, M., Kim, T.-H., Heffron, J., and White, J. (2018). Game based Cybersecurity Training for High School Students. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pages 68–73, Baltimore Maryland USA. ACM.
- Komodo Research (2019). One Push Too Far / Part 3: Notification-Based Botnet Risks.

- Krebs, B. (2020). Be Very Sparing in Allowing Site Notifications.
- Lee, J., Kim, H., Park, J., Shin, I., and Son, S. (2018). Pride and Prejudice in Progressive Web Apps: Abusing Native App-like Features in Web Applications. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS '18*, pages 1731–1746, New York, NY, USA. Association for Computing Machinery.
- Li, Z. (2018). NPS® question is available in Microsoft Forms.
- Lloyds (2024). Consumer Digital Index. Technical report.
- Lo, V. (2021). Google Chrome Notification Analysis in Depth. Technical report, The SANS Institute, SANS Institute 11200 Rockville Pike, Suite 200 North Bethesda, MD 20852.
- Malwarebytes (2022). PUP.Optional.PushNotifications.
- Mancino, D. (2023). Digital literacy in the EU: An overview.
- Matsuzaki, R. and Sato, M. (2025). Detecting Phishing-Targeted Web Push Notifications Through Image Similarity Analysis. In Lee, J.-H., Emura, K., and Lee, S., editors, *Information Security Applications*, pages 257–269, Singapore. Springer Nature.
- MDN (2023a). How to make PWAs re-engageable using Notifications and Push.
- MDN (2023b). Web Push API Notifications best practices.
- MDN (2024a). Notifications API.
- MDN (2024b). Push API.
- Nielsen, J. (2024). 10 Usability Heuristics for User Interface Design.
- Norman, D. (2013). *The Design of Everyday Things: Revised and Expanded Edition*. Hachette UK. Google-Books-ID: I1o4DgAAQBAJ.
- ONS (2019). Exploring the UK’s digital divide.
- ONS (2024). User guide to crime statistics for England and Wales.

OWASP (2025). Password Storage Cheat Sheet.

Pan, Y., White, J., and Sun, Y. (2016). Assessing the threat of web worker distributed attacks. In *2016 IEEE Conference on Communications and Network Security (CNS)*, pages 306–314.

Pantelakis, G., Papadopoulos, P., Kourtellis, N., and Markatos, E. P. (2022). Measuring the (Over)use of Service Workers for In-Page Push Advertising Purposes. In Hohlfeld, O., Moura, G., and Pelsser, C., editors, *Passive and Active Measurement*, pages 426–438, Cham. Springer International Publishing.

Papadopoulos, P., Ilia, P., Polychronakis, M., Markatos, E. P., Ioannidis, S., and Vasiladiadis, G. (2019). Master of Web Puppets: Abusing Web Browsers for Persistent and Stealthy Computation. In *Proceedings 2019 Network and Distributed System Security Symposium*, San Diego, CA. Internet Society.

Richards, M., Kravitz, L., Cropp, R., and Song, J. (2020). Reducing distractions with quiet notification requests.

Rushanan, M., Russell, D., and Rubin, A. D. (2016). MalloryWorker: Stealthy Computation and Covert Channels Using Web Workers. In Barthe, G., Markatos, E., and Samarati, P., editors, *Security and Trust Management*, pages 196–211, Cham. Springer International Publishing.

Samaana, H., Costa, D. E., Abdellatif, A., and Shihab, E. (2025). Opportunities and security risks of technical leverage: A replication study on the NPM ecosystem. *Empirical Software Engineering*, 30(3):96.

Sato, M. (2025). Re: Please could you help me understand your paper "Control Displaying of Web Push Notifications Inducing Users to Phishing Websites".

Sato, M. and Mandai, R. (2024). Control Displaying of Web Push Notifications Inducing Users to Phishing Websites. In *2024 IEEE Conference on Dependable and Secure Computing (DSC)*, pages 102–103.

Schneier, B. (2013). On Security Awareness Training.

Sommerville, I. (2015). *Software Engineering, Global Edition*. Pearson Education, Limited, Harlow, UNITED KINGDOM.

- Steventon-Barnes, Z. (2024). Declaring Total War On The (Web Notifications) Pusherman: Educating People About The Risks Of Web Push Notification Scams - Progress Report.
- Subramani, K., Jueckstock, J., Kapravelos, A., and Perdisci, R. (2022). SoK: Worker-ounds - Categorizing Service Worker Attacks and Mitigations. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 555–571, Genoa, Italy. IEEE.
- Subramani, K., Yuan, X., Setayeshfar, O., Vadrevu, P., Lee, K. H., and Perdisci, R. (2020). When Push Comes to Ads: Measuring the Rise of (Malicious) Push Advertising. In *Proceedings of the ACM Internet Measurement Conference*, pages 724–737, Virtual Event USA. ACM.
- Tetri, P. and Vuorinen, J. (2013). Dissecting social engineering. *Behaviour & Information Technology*, 32(10):1014–1023.
- Ubah, K. (2024). CommonJS vs. ES modules in Node.js.
- Vanyashin, A. (2016). README.md.
- W3C (2024). Push API.
- WAI (2024). Introduction to Understanding WCAG 2.2.
- Wall, D. (2007). *Cybercrime: The Transformation of Crime in the Information Age*. Polity.
- Wall, D. S. (2024). *Cybercrime: The Transformation of Crime in the Information Age*. John Wiley & Sons. Google-Books-ID: DIMCEQAAQBAJ.
- Wang, Z., Sun, L., and Zhu, H. (2020). Defining Social Engineering in Cybersecurity. *IEEE Access*, 8:85094–85115. Conference Name: IEEE Access.

A. Graphs for survey results

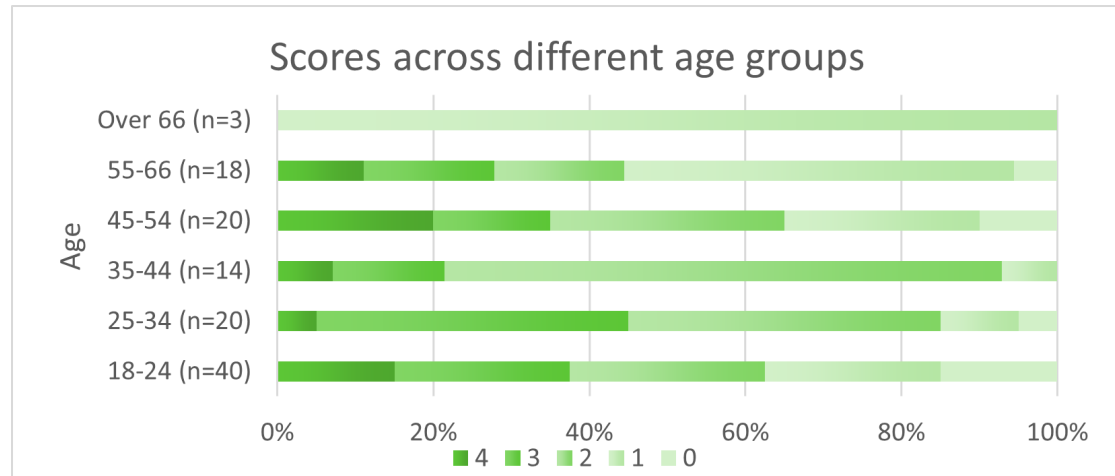


Figure 8: Proportion of respondents attaining a given score (number of correct responses to questions 1 - 4) in the survey by age group and the number of people (n) from that age group.

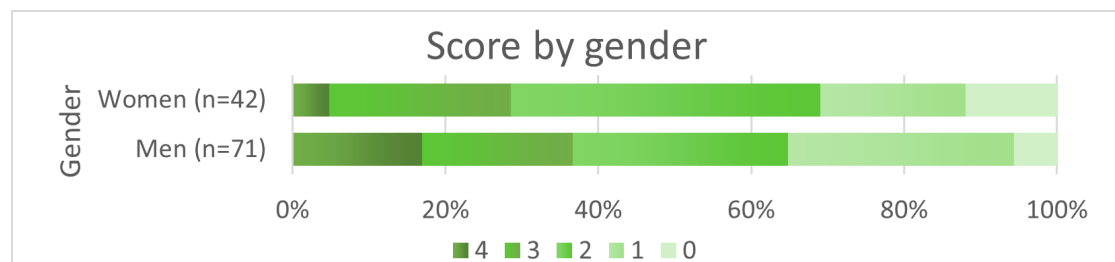


Figure 9: Proportion of respondents attaining a given score (number of correct responses to questions 1 - 4) in the survey by gender and the number of people (n) in that gender. The number of respondents that responded “other” or “prefer not to say” were too small to produce reliable data.

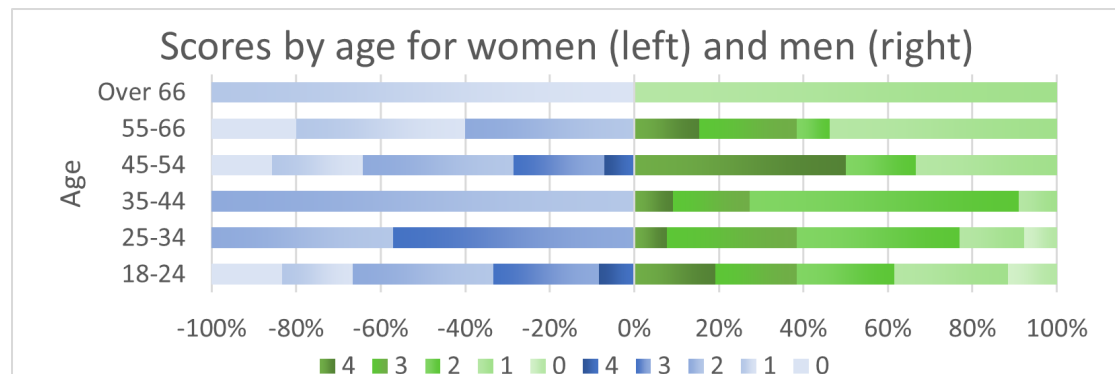


Figure 10: Proportion of respondents attaining a given score (number of correct responses to questions 1 - 4) in the survey by age group subdivided by gender (women left in blue and men right in green). The number of respondents that responded to the gender question with "other" or "prefer not to say" were too small to produce reliable data.

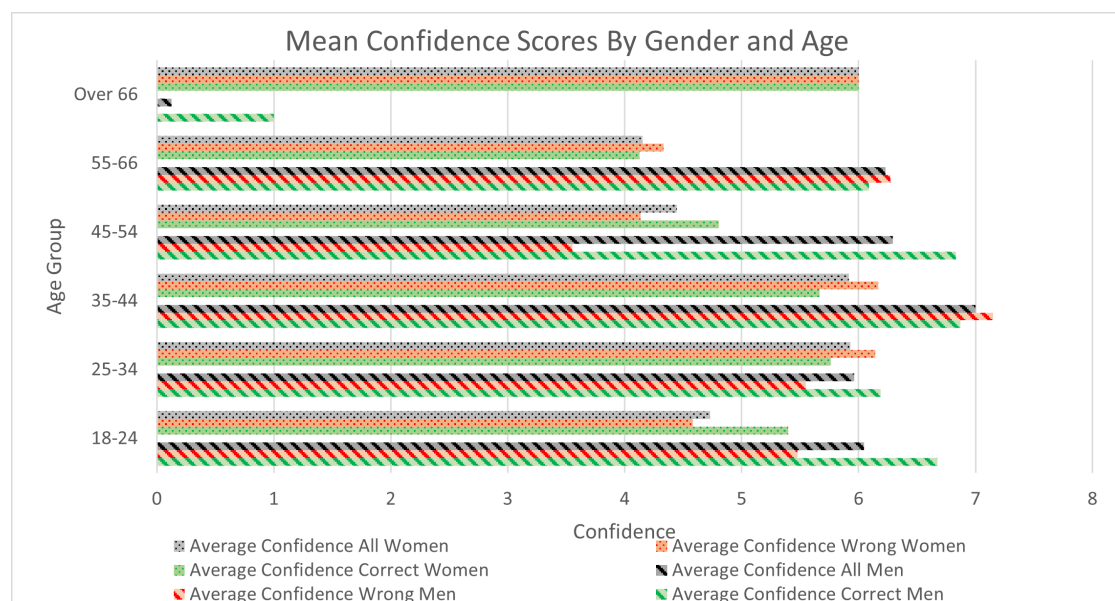


Figure 11: Mean confidence scores on questions 1 - 4 divided by age group (left), gender (men striped, women spotted), and whether the answer was correct or wrong (green correct, red wrong, all answers in grey).

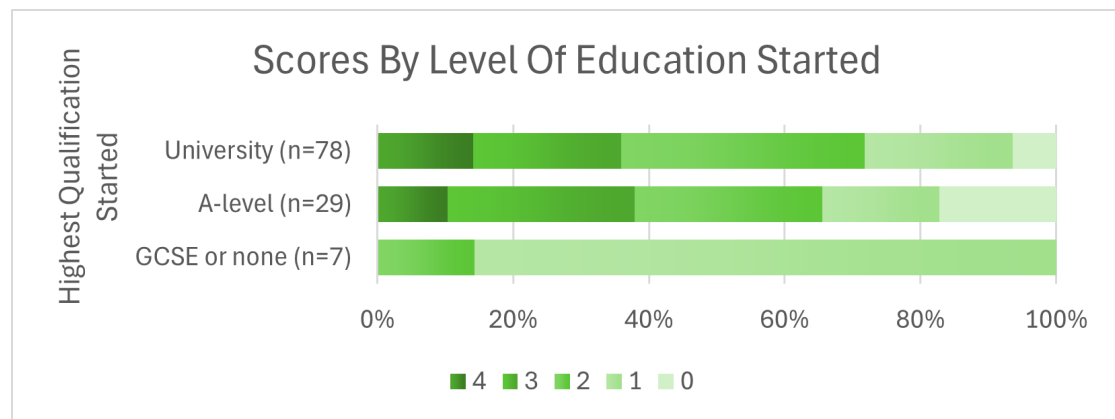


Figure 12: Proportion of respondents attaining a given score (number of correct responses to questions 1 - 4) in the survey by level of educational qualification started and the number of people (n) who started that qualification.

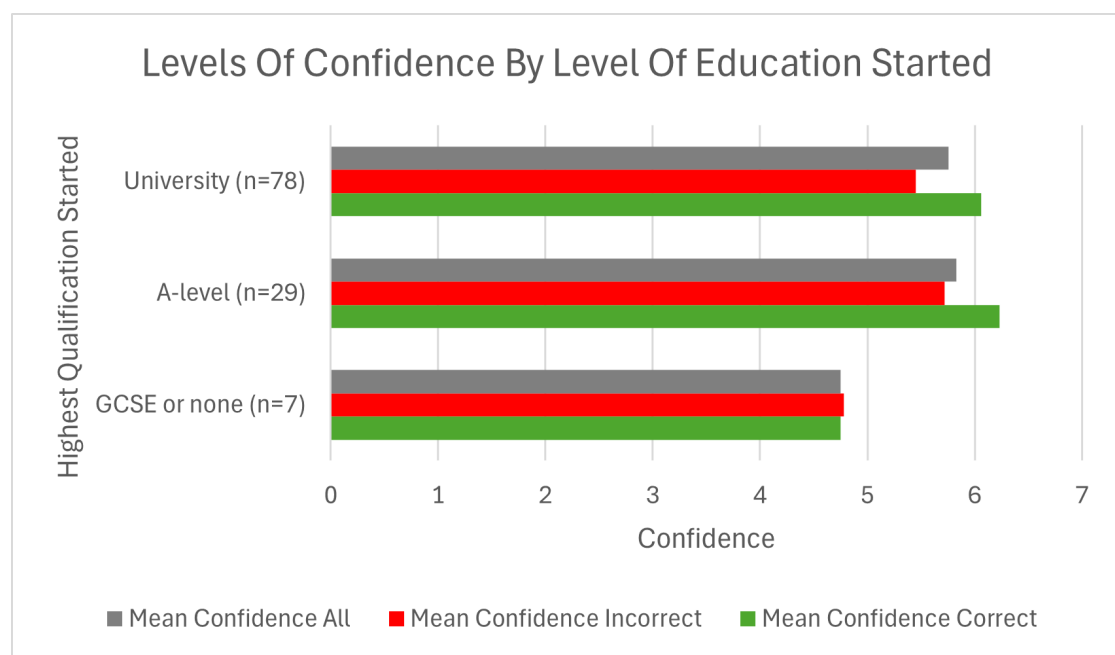


Figure 13: Mean confidence scores on questions 1 - 4 divided by the level of education started (left) and whether the answer was correct (green) or incorrect (red) with the mean for all answers in grey and the number of people (n) who started that qualification.

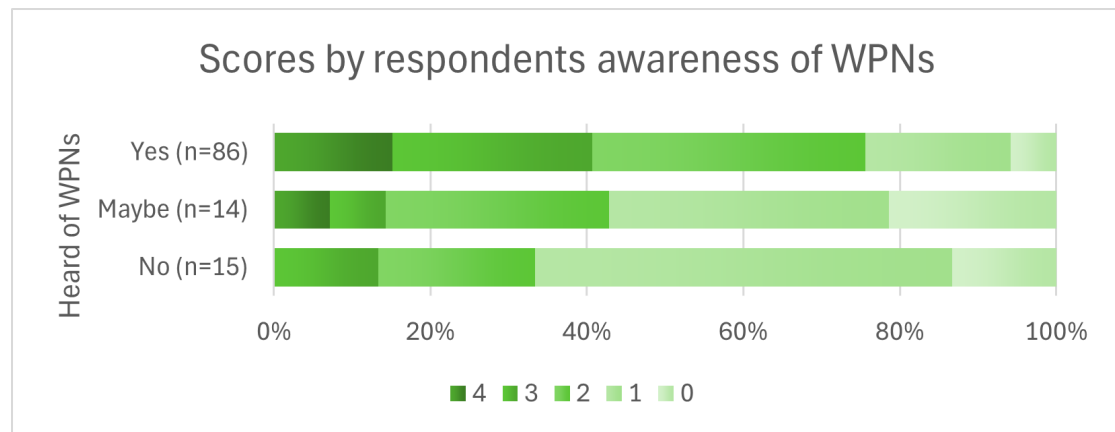


Figure 14: Proportion of respondents attaining a given score (number of correct responses to questions 1 - 4) in the survey by their response to the question “Have you previously heard the term ‘Web Push Notifications?’” and the number of people (n) within that group.

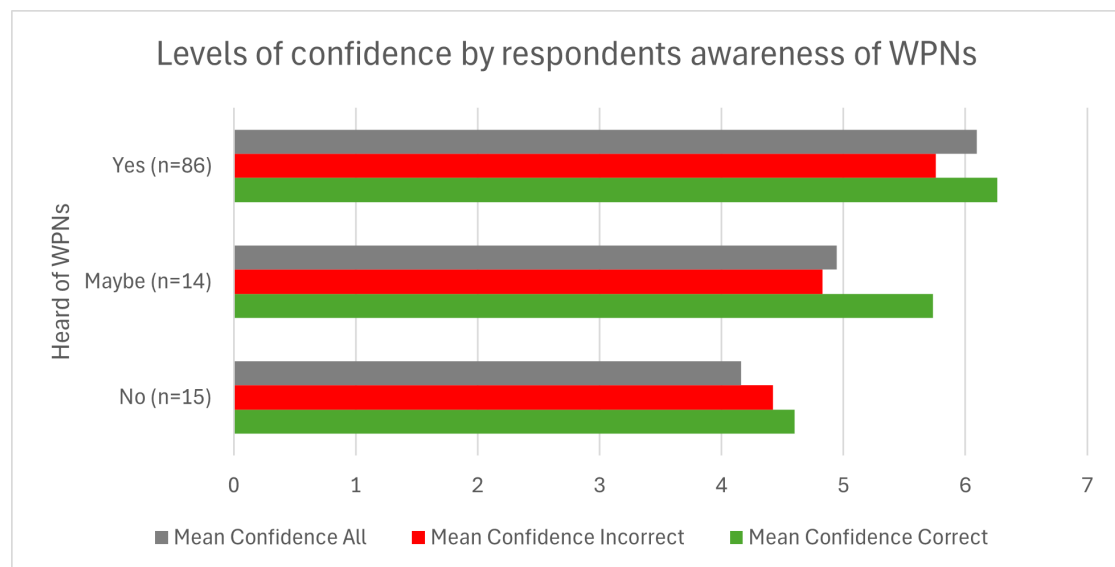


Figure 15: Mean confidence scores on questions 1 - 4 divided by response to the question “Have you previously heard the term ‘Web Push Notifications?’” and whether the answer was correct (green) or incorrect (red) with the mean for all answers in grey and the number of people (n) within that group.

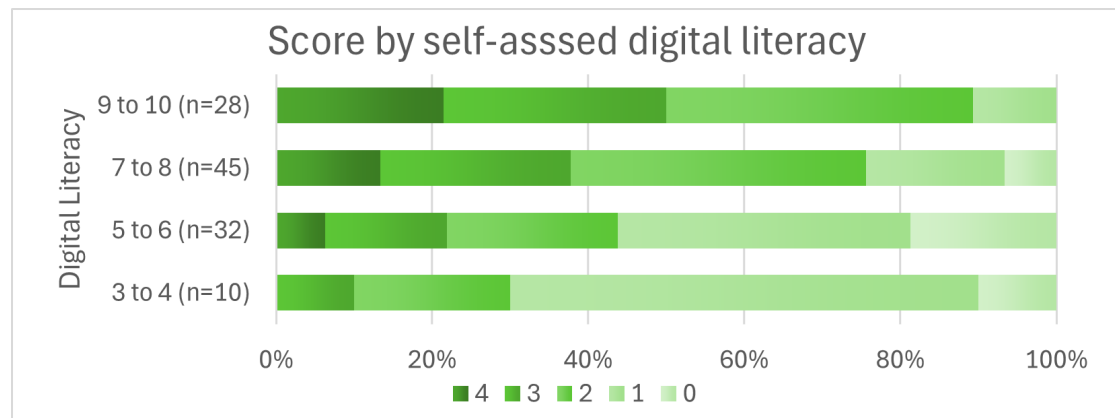


Figure 16: Proportion of respondents attaining a given score (number of correct responses to questions 1 - 4) in the survey by self-assessed digital literacy and the number of people (n) within that group.

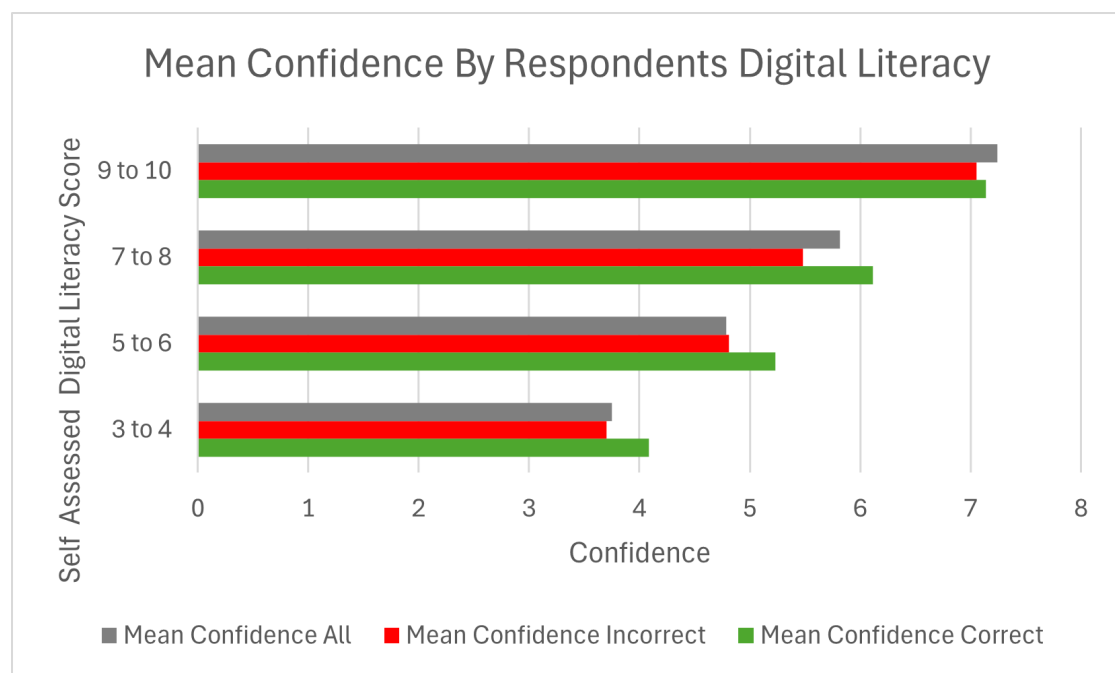


Figure 17: Mean confidence scores on questions 1 - 4 divided by self-assessed digital literacy and whether the answer was correct (green) or incorrect (red) with the mean for all answers in grey and the number of people (n) within that group.

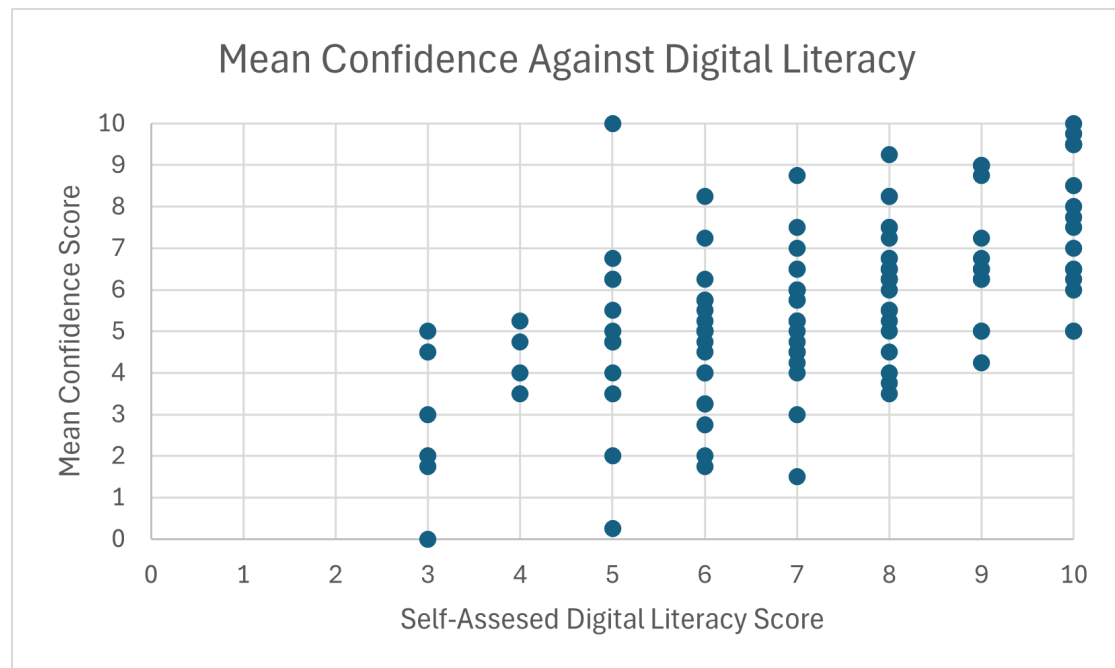


Figure 18: Mean confidence scores on questions 1 - 4 (y-axis) against their self-assessed digital literacy (x-axis).

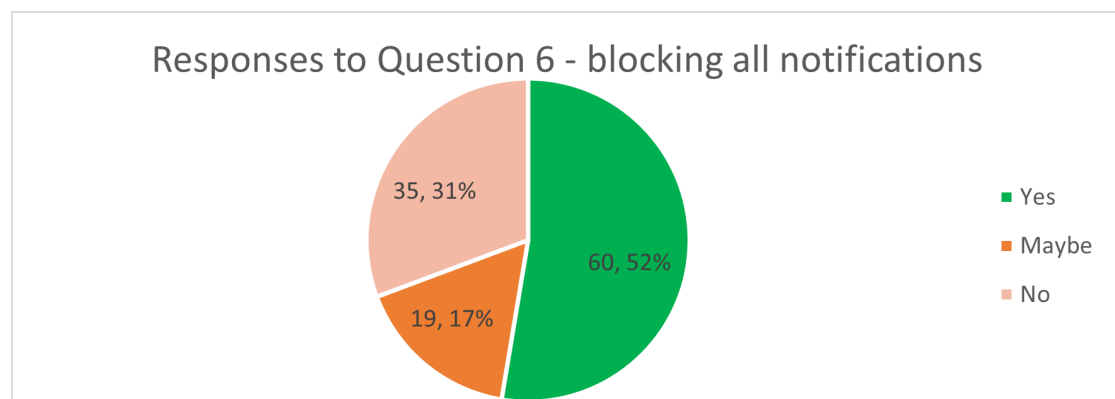


Figure 19: Responses (number and percentage) to question 6 - “Do you know how to disable all notifications from websites in your web browser?” The correct answer is shown in green.

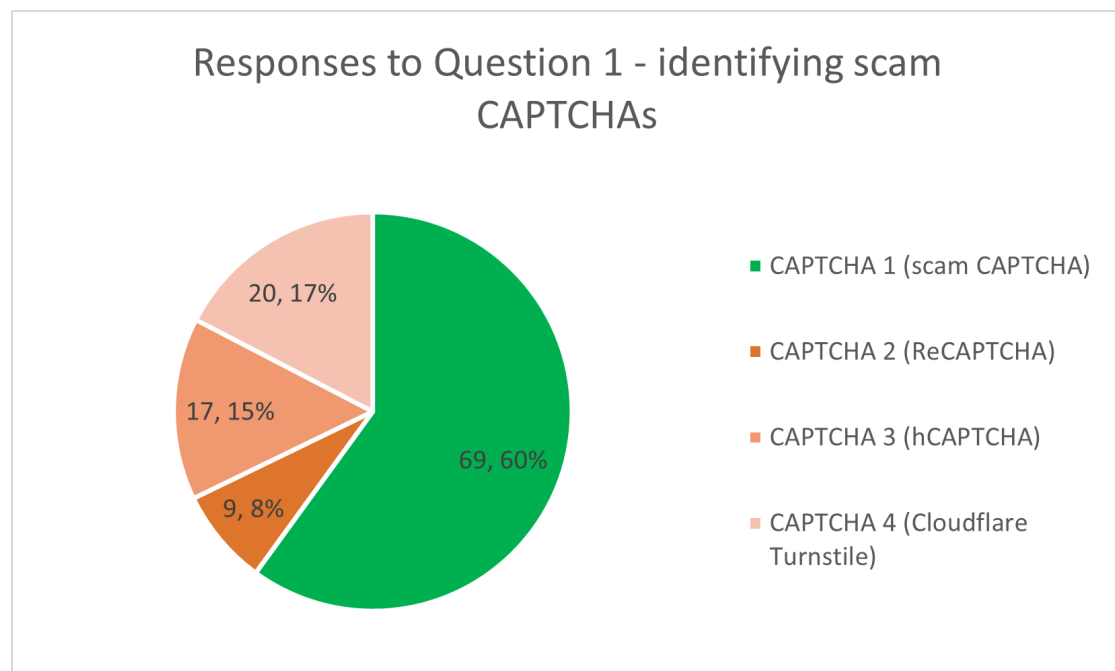


Figure 20: Responses (number and percentage) to question 1 - “Websites sometimes use CAPTCHAs to check you are a human, however scammers sometimes use fake CAPTCHAs to trick users. Which of the following images do you think is a fake CAPTCHA?” The correct answer is shown in green.

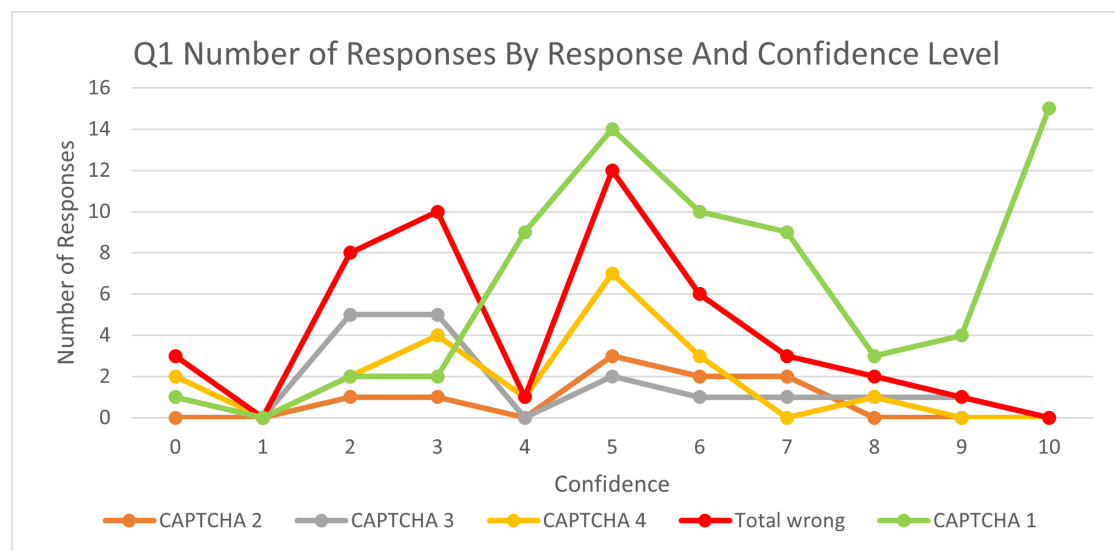


Figure 21: Number of responses (y-axis) by level of confidence (x-axis) for different answers to question 1.

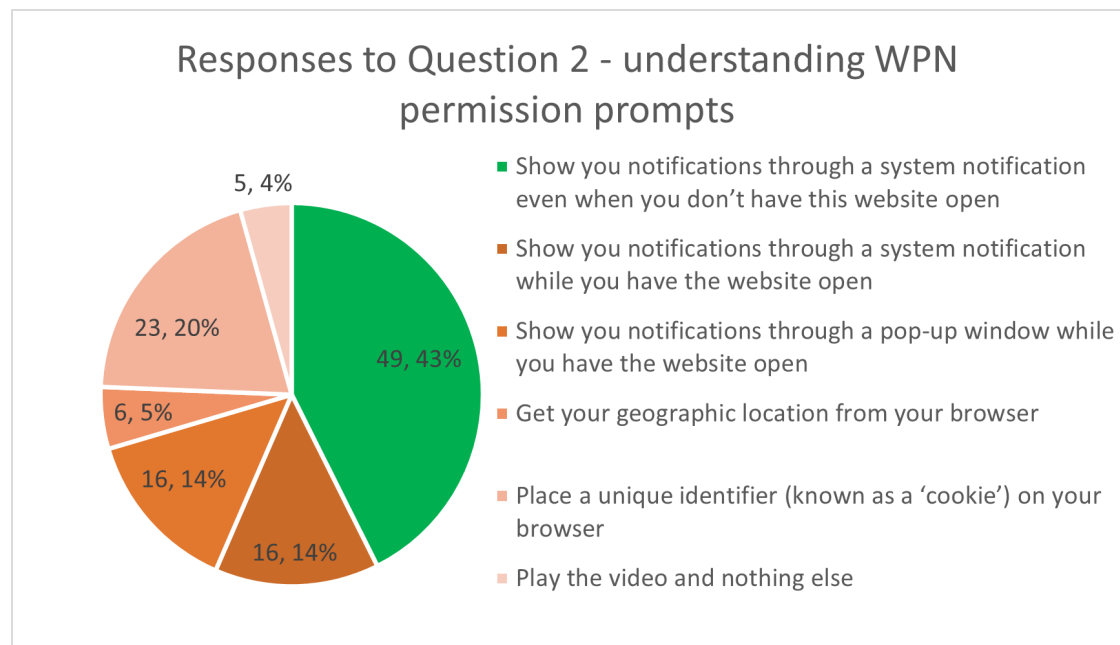


Figure 22: Responses (number and percentage) to question 2 - “Imagine you have gone to the website pictured above to watch a video. Which of the following do you think clicking allow would do?” The correct answer is shown in green. The description that “a system notification *outside the browser (such as the Windows Notification Centre)*” has been omitted from answer names.

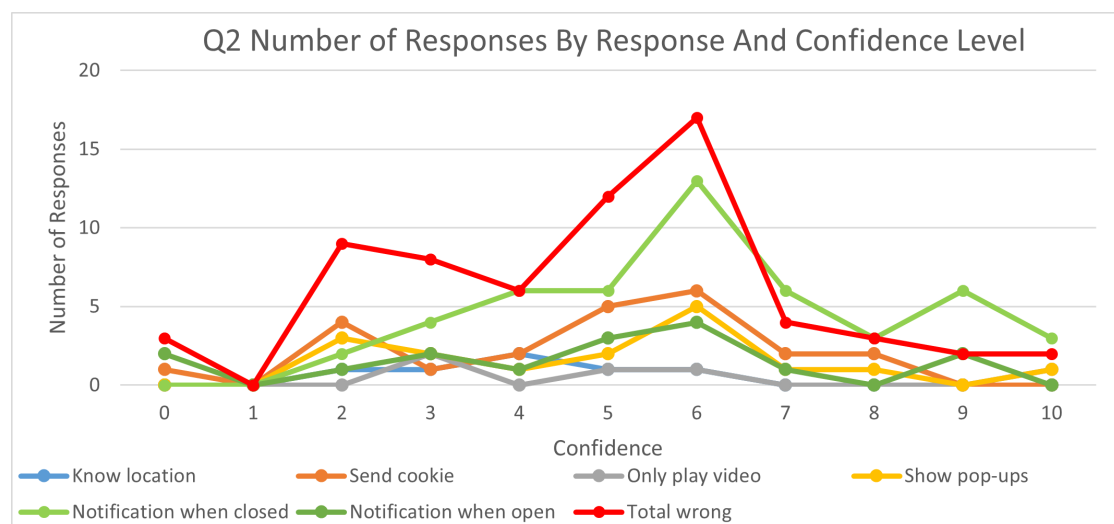


Figure 23: Number of responses (y-axis) by level of confidence (x-axis) for different answers to question 2.

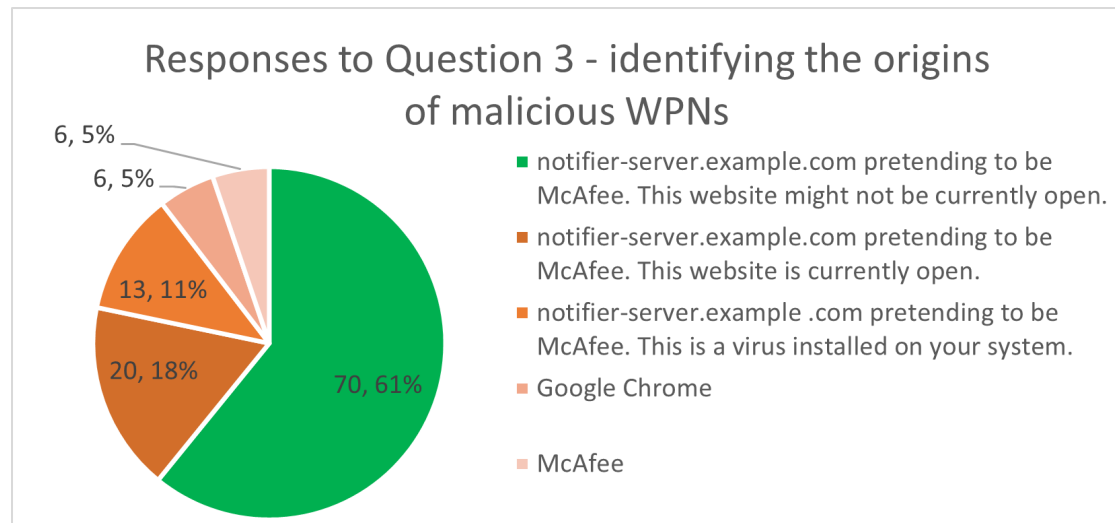


Figure 24: Responses (number and percentage) to question 3 - “Imagine you have gone to the website pictured above to watch a video. Which of the following do you think clicking allow would do?” The correct answer is shown in green. The description that “*This website is sending the notification through Google Chrome*” has been omitted from answer names.

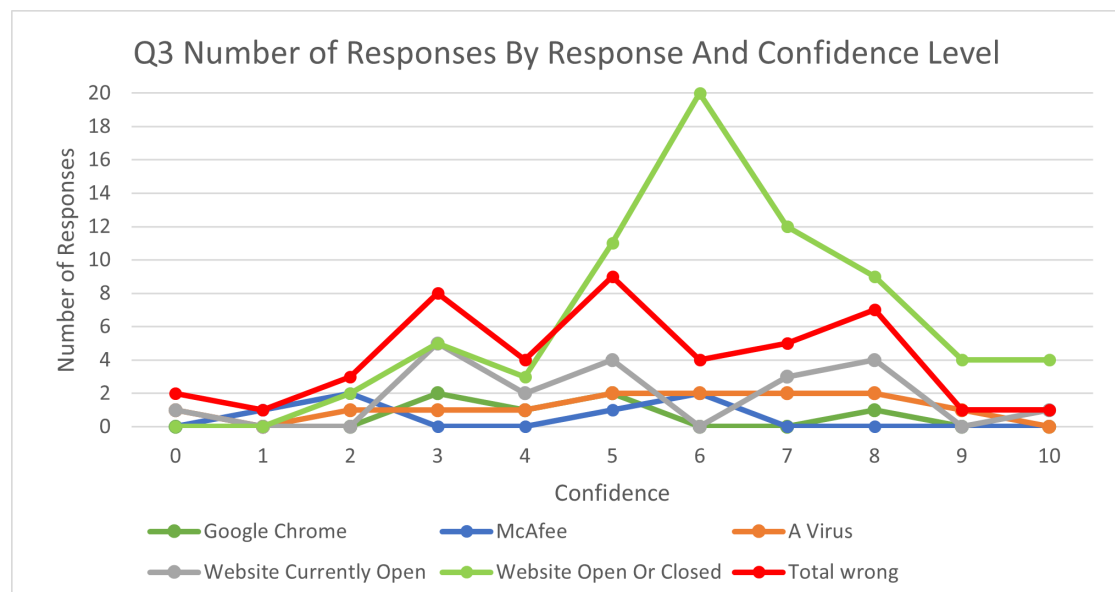


Figure 25: Number of responses (y-axis) by level of confidence (x-axis) for different answers to question 3.

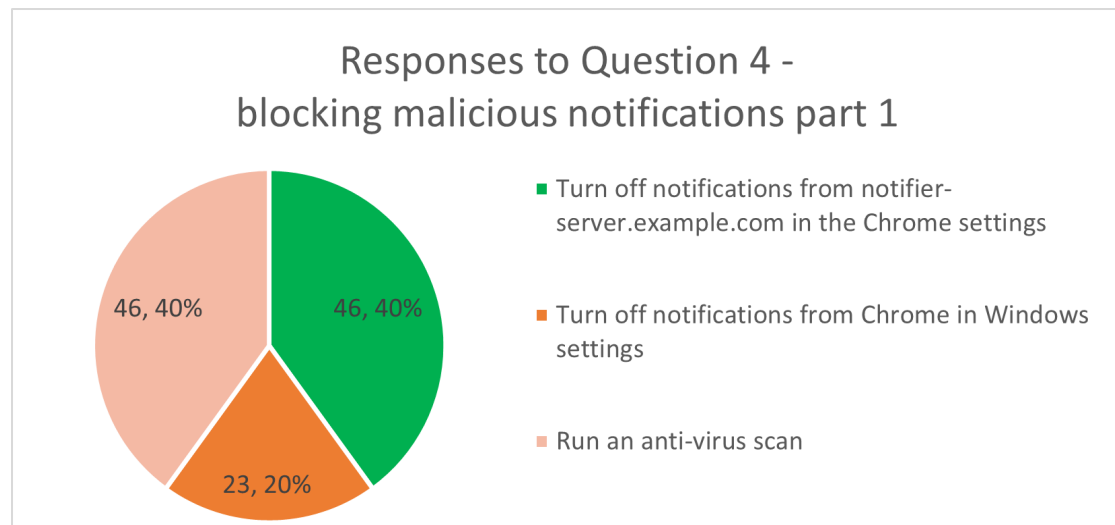


Figure 26: Responses (number and percentage) to question 4 - “Pictured is an example of a scam notification. How do you think you could prevent further scam notifications like this from appearing?” The correct answer is shown in green. Notes instructing users to replace Chrome and Windows with the software they use has been omitted from answer names.

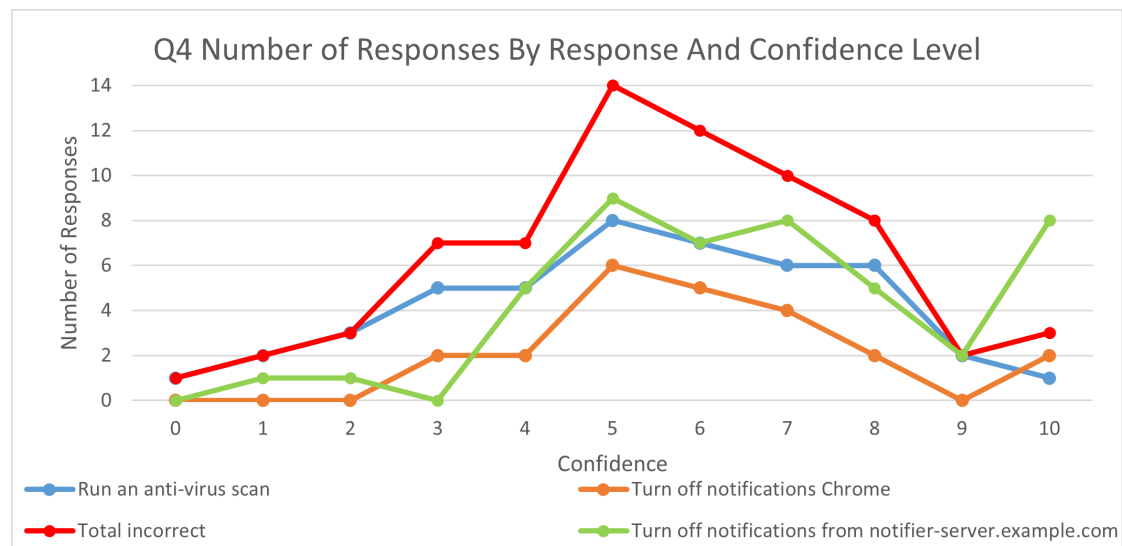


Figure 27: Number of responses (y-axis) by level of confidence (x-axis) for different answers to question 4.

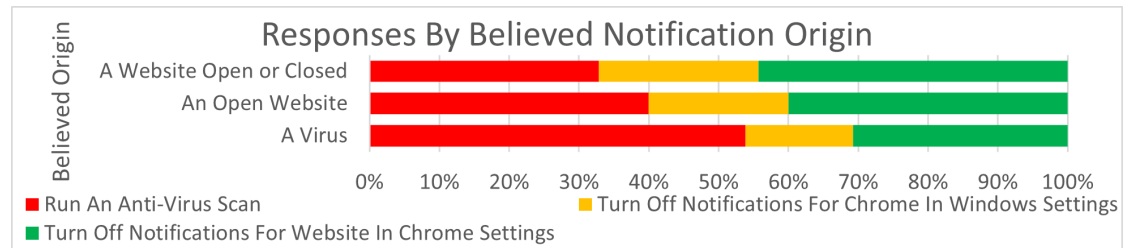


Figure 28: Participants' approach to dealing with a malicious WPN by their belief of the origin of the WPN. I have excluded respondents who did not believe that the notification was malicious in their response to question 3.

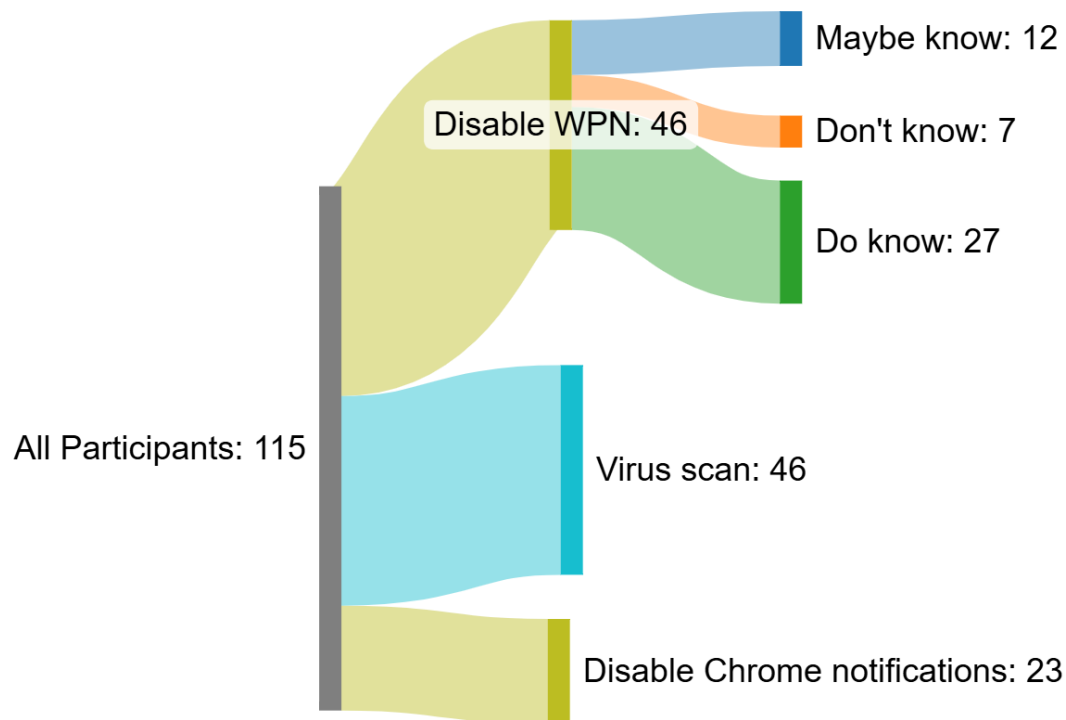


Figure 29: A Sankey diagram showing respondents approach to dealing with a malicious WPN and for the best approach whether or not they know how to accomplish it.

B. Web Push Notifications Network Diagram

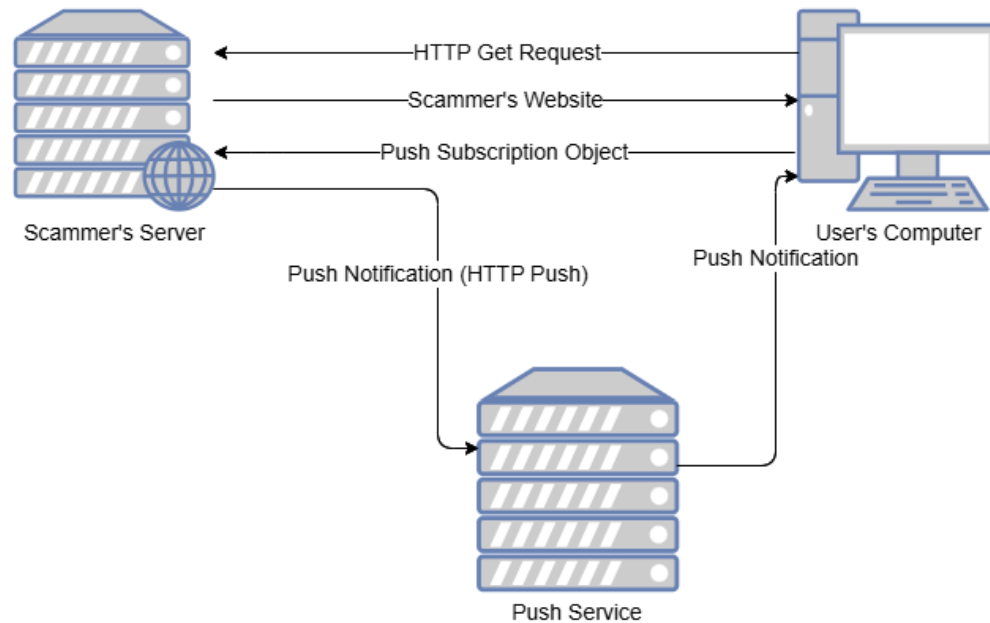


Figure 30: Web Push Notification network diagram

C. Web Push Notifications Sequence Diagram

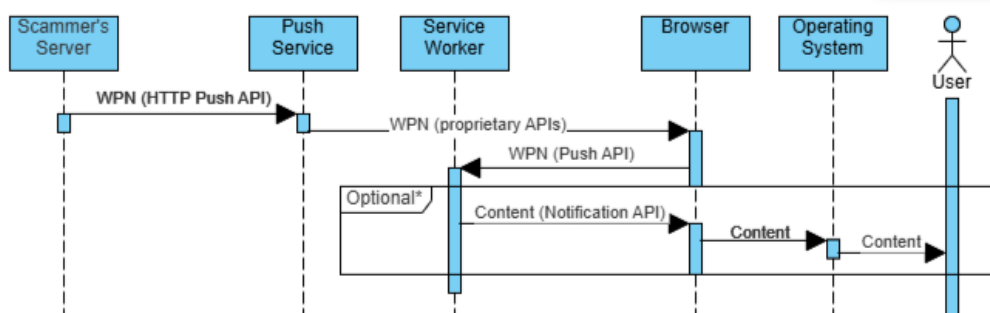


Figure 31: Sequence diagram for sending a WPN.

* = this step can be optional depending on the web browser.

D. Mid-Fi Prototype Of Homepage

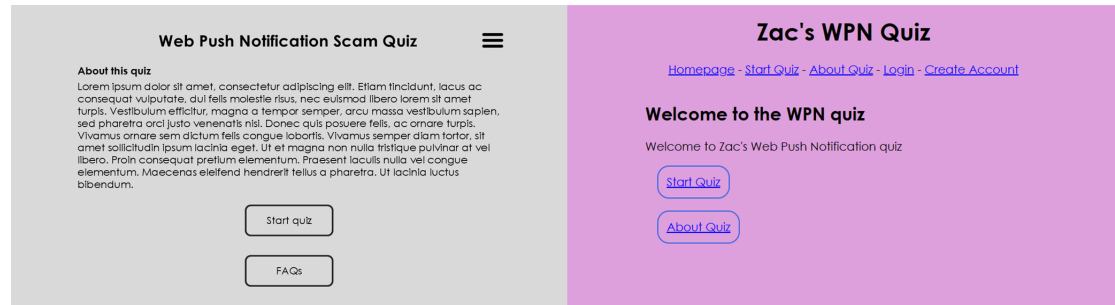


Figure 32: A mid-fi prototype of the homepage (left) and the finished product (right)

E. Mid-Fi Prototype Of Question

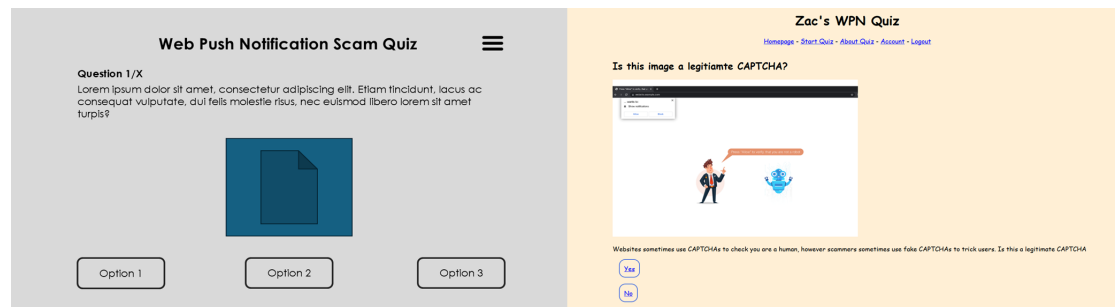


Figure 33: A mid-fi prototype of an example question (left) and the finished product with the background colour and font changed (right)

F. Textual Use Case Question

Table 1: Textual Use Case For Answer Question

Use case name:	Answer Question
Version:	1
Primary Actor:	<i>Quiz User</i>
Secondary Actors:	None
Goal:	<i>Quiz User</i> learns the correct answer to the question in an enjoyable manner.
Summary:	<i>Quiz User</i> answers a question that is part of the quiz and is told whether their answer is correct or not.
Trigger:	If this is the first question, <i>Quiz User</i> clicks the start quiz button. Otherwise, <i>Quiz User</i> finishes the previous question.
Main success scenario:	1. <i>Quiz User</i> views the question 2. <i>Quiz User</i> selects the correct answer to the question 3. The quiz tells <i>Quiz User</i> they are correct
Alternatives:	2a.1 <i>Quiz User</i> selects an incorrect answer to the question 2a.2 The quiz tells <i>Quiz User</i> their answer is incorrect and which the correct answer is.
Exceptions:	None
Postconditions:	<i>Quiz User</i> moves on to the next question

G. Flow Chart Question

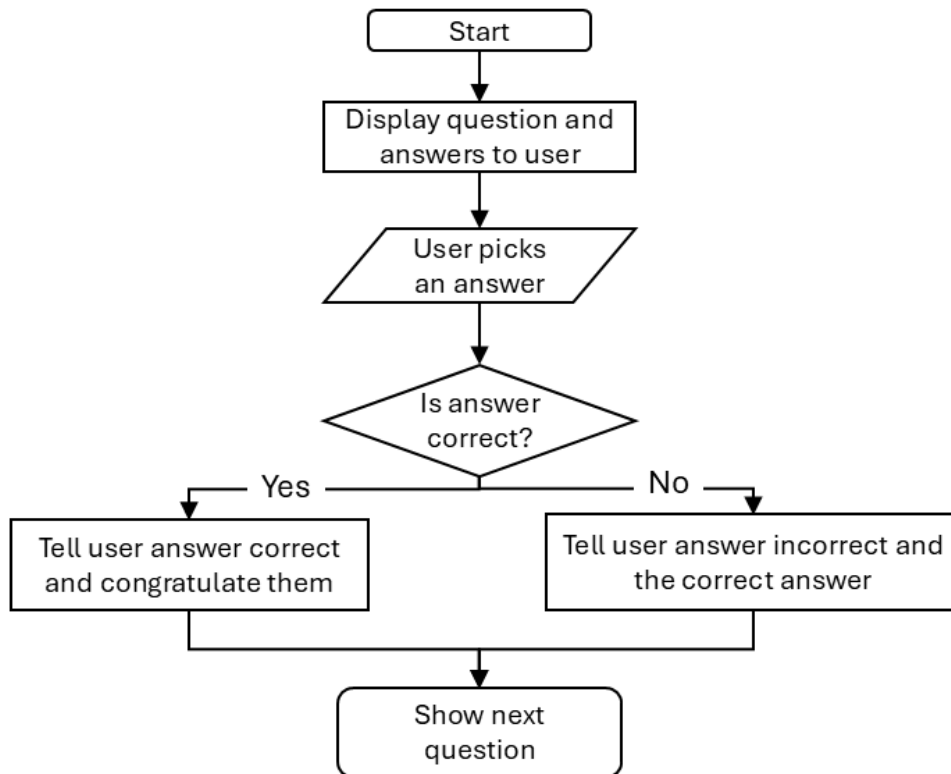


Figure 34: A flow chart showing the process of a user answering a question

H. Dependencies

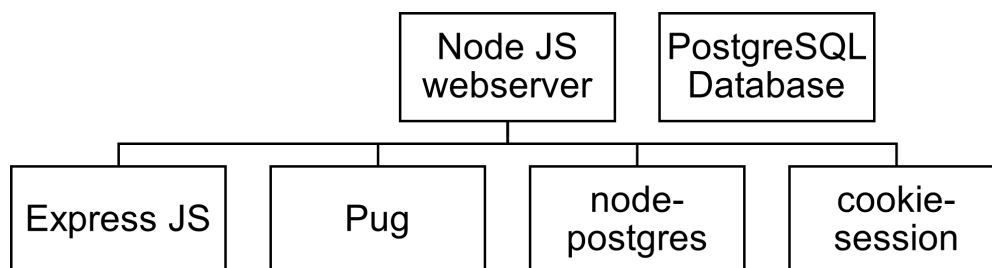


Figure 35: A diagram of the dependencies the system uses

I. Evaluation against MoSCoW requirements titles

Table 2: Status of MoSCoW requirements - Implemented (I) or Not Implemented (NI)

Requirement number:	Results
1 - Allow users to answer multi-choice questions	I
2 - Show users text for questions	I
3 - Tell users if their answer was correct	I
4 - Give users question specific feedback	I
5 - Show users images for questions	I
6 - Be compliant with WCAG 2.2 at AA level	NI - Impossible, see Section 6.8
7 - Allow users to view previous scores	I
8 - Show users example web push notifications	NI – insufficient time ^a
9 - Allow users to change settings to better suit visual impairments and dyslexia	I
10 - Allow questions to be edited by editing a JSON file	I
11 - Allow images to be zoomed in on	NI – insufficient time.
12 - A mechanism to save user progress	This is technically implemented ^b
13 - A mechanism to send questions specific to the browser	NI

^a especially as WPNs can only be sent from a website with HTTPS support.

^b if a user browses back to the last question they left off at, the attempt JSON will still have their current score. However the quiz homepage is not set up to direct users to resume an attempt.

J. Database Excerpt

	username [PK] character varying (50)	password character varying (200)	salt character varying (200)	settings jsonb
1	Arnold Aardvark	hPHZ/nolGJeXM1gXBK...	i+Lq+OxWNx+gG69ikSvKw...	{ "font": 3, "username": "Arnold Aardvark", "background": "purple", "individualUser": tr...

Figure 36: An excerpt from the database, note the apparently random value in the hash field that bears no correspondence to the password the user entered.

K. Password Hashing Method

```
#hashPassword (password, salt){
  const SCRYPT_COST_EXPONENT = 15
  const SCRYPT_BLOCK_SIZE = 8
  const SCRYPT_PARALLELIZATION = 1
  //https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Password_Storage_Cheat_Sheet.md
  //upgrade when it becomes appropriate

  return scryptSync(password, salt, 64, { N: 2 ** SCRYPT_COST_EXPONENT,
    r: SCRYPT_BLOCK_SIZE,
    p: SCRYPT_PARALLELIZATION,
    maxmem: 130 * (2 ** SCRYPT_COST_EXPONENT) * SCRYPT_BLOCK_SIZE });
}
```

Figure 37: The method used to hash, salt, and stretch passwords.

L. Pug Question Block

```
h1= data.title
img(src=data.image class="question-image")
p= data.text
```

Figure 38: The Pug block used to display questions. Note the use of the = sign after the tag name as this prompts Pug to perform XSS sanitation.

M. Testing Results From Question 5

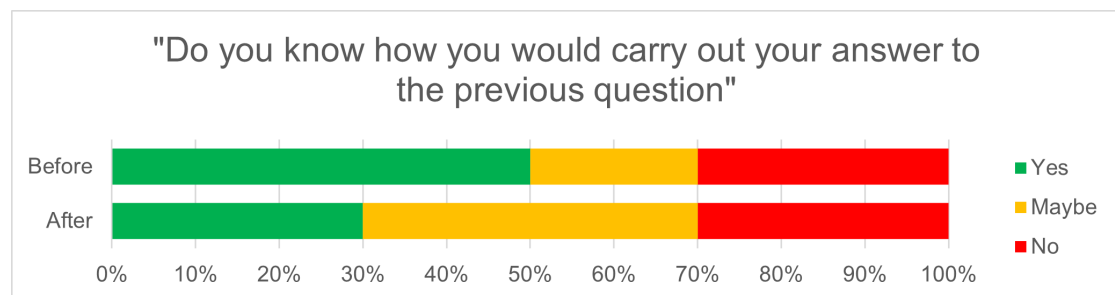


Figure 39: Results from question 5 before and after using the educational quiz website.

P. Positive Qualitative Feedback



Figure 42: A word cloud generated by Forms for what users liked about the system.

Q. Negative Qualitative Feedback



Figure 43: A word cloud generated by Forms for what users disliked about the system.

R. Glossary

Application Programming Interface (API) A standardised mechanism for one program to communicate with another, such as a website with a browser

Botnet A network of zombified computers which may be used for cryptocurrency mining, password cracking, DDoS attacks etc.

Cross Site Request Forgery (XSRF) Causing a legitimate user to send a malicious request to a legitimate website.

Cross Site Scripting (XSS) The insertion of malicious JavaScript into a legitimate web-site.

ECMAScript Modules A standardised JavaScript mechanism to call functions or use objects in one file from another.

HTTP Push A standard to connect to Push Services maintained by the IETF.

Notifications API A standard for a service worker to display a notification (including WPNs) to the user maintained by the WHATWG.

NPM Libraries Libraries distributed using NPM

Push API A standard for the browser to pass WPNs to a service worker maintained by the W3C.

Push Services A mechanism for delivering WPNs to the browser. They are chosen by the browser.

Service Worker A piece of JavaScript code that runs on the user's browser even when its website is closed.

Social Engineering Defined by Hadnagy (2018) as “any act that influences a person to take an action that may or may not be in his or her best interests”

System Notification A notification displayed through APIs provided by the operating system (Bilogrevic et al., 2021). It generally appears momentarily on the user's screen and can then be found in a dedicated area of the Operating System's GUI

Unified Modelling Language (UML) A set of diagrams commonly used to model computer systems.

User Experience (UX) The way in which a user interacts with a piece of software.

Web Push Notification (WPN) A notification sent to a user through a combination of the Push API, HTTP Push, and the Notifications API.

Zombie, Zombified A computer owned by a legitimate user but being utilised by a criminal as part of a botnet.